Open-ended Spoken Language Technology: Studies on Spoken Dialogue Systems and Spoken Document Retrieval Systems

Naoyuki Kanda

Abstract

Spoken language is one of the most natural communication tools; therefore, technologies for handling spoken language by using computers have been attracting much attention. One example of such technology is a speech interface used as a powerful man-machine interface, and another is a speech mining system used as a rich information extraction tool. With advances in speech recognition techniques, some systems have reached the level of practicality, and there is a growing demand for applying spoken language technologies anywhere for any purpose. However, most spoken language systems, especially application systems such as spoken dialogue and spoken document retrieval systems, still depend on limited domain knowledge, and what the system can interpret is severely limited. Since spoken language is so familiar in our daily lives, users often expect spoken language systems to understand everything the users can. Therefore, there is a large gap between what current systems can handle and what users expect. It is important to narrow this gap in spoken language technologies to benefit a wide range of users.

Our ultimate goal is to develop open-ended spoken language technologies, which can handle a broad range of utterances by freely extending their knowledge resources. For achieving this goal, we tackled two technical problems; (i) spoken language analysis with little or no domain knowledge as a basis for handling arbitrary topics and (ii) integration of knowledge on an arbitrary number of domains. For spoken language systems, speech recognition errors are inevitable and need to be considered in each layer of a system. Therefore, to solve these problems, we investigated two target systems, spoken dialogue and spoken document retrieval. A spoken dialogue system is a computer system that can understand and respond to a user's spoken requests. We consider it as a representative system that treats speech as a man-machine interface. A spoken document retrieval system can find specific recordings from input queries. We consider it as a representative system that treats speech as an information source. We developed open-ended technologies for both systems by solving problems (i) and (ii).

Chapter 1 describes the background of this study and Chapter 2 then overviews related studies and describes the position of the thesis.

Chapter 3 proposes a robust language understanding method for dialogue systems on the database search task as a solution to problem (i) for spoken dialogue systems. The proposed method can be used even when the database is not the one that the system is prepared for. To develop such a language-understanding method, we also propose general context models for the database search task and incorporate context information from these models into the language-understanding procedure.

Chapter 4 proposes a spoken dialogue system that consists of multiple subsystems on different domains and can easily extend new domain knowledge as a solution to problem (ii) for spoken dialogue systems. The key component of this system is a domain selection method, which select an appropriate domain for each user utterance. We proposed a new domain selection scheme to determine whether the previously-domain should be kept or not. Experimental results indicate that this domain selection method achieved better results than conventional methods with achieving the ability to extend new domain knowledge.

Chapter 5 proposes a spoken document retrieval system with new indexing and searching methods that can detect positions of arbitrary keywords from speech database as a solution to problem (i). To achieve fast and accurate term detection, we tandemly combined three types of open vocabulary indexing methods. Experimental results indicated that the proposed method clearly outperformed conventional spoken term detection methods, especially when searching for out-of-vocabulary keywords.

Chapter 6 proposes an index combination method for spoken term detection systems as a solution to problem (ii). Simply combining many indices increases index size, which raises the problems of high storage cost and slow search speed. To suppress the increase in index size, we proposed a selective index method based on an out-of-vocabulary region classifier. The selective index combination method can suppress this increase while improving search accuracy.

Finally, in Chapter 7 concludes the thesis with a discussion on the contributions of our study and future directions.

論文梗概

音声は人間が用いる最も自然なコミュニケーション手段の１つである．そのため音声を計算機で扱うことで，携帯電話やカーナビゲーションシステムの操作といった自然で強力なマンマシンインタフェースや，コールセンター通話録音に基づく評判分析のような，音声を豊富な情報源と見立てた情報抽出システムが実現されると期待されている．近年に入り音声認識技術が大きく発展し，一部の技術は既に実用の領域に達している．一方で従来の音声言語処理システムの多くは限られたドメイン知識に依存しており，結果としてそのシステムが処理できる音声の内容が強く限定されている．特に音声対話システムや音声文書検索システムなど，音声認識の結果を応用するシステムにおいてその問題が顕著に現れる．これらのシステムのユーザにとって音声は自然に利用できるものであるが故に，システムが処理できる音声とユーザが期待するものとの乖離は大きく，この解決は音声言語処理がさらに広く利用されるために重要である．

本研究の究極的な目標は，広範な種類の音声を処理でき，新規知識の自由な拡張も可能な音声言語処理技術の開発である．本論文ではこのために，(i) ドメイン知識が少ないもしくは無い状態でも頑健に動作する音声言語の解析技術と (ii) 新規ドメイン知識の自由な拡張が可能なシステム統合技術の開発に取り組んだ．これらの課題は，一部のモジュールではなくシステム全体として考慮する必要がある．そのため本論文では，音声対話システムと音声文書検索システムという代表的な 2 つの音声言語処理システムを構築し，それぞれにおいて上記 (i)(ii) の課題に取り組んだ．音声対話システムは，計算機がユーザの音声を認識，理解し適切な応答を返すことが可能なシステムであり，音声をインタフェースとみなす技術の題材として選択した．一方で音声文書検索システムとは，音声データを情報を蓄えた文書（音声文書）とみなし，音声文書から効率的かつ高精度に情報を抽出するための技術であり，音声を情報源とみなす技術の題材として選択した．本論文では各システムにおいて上記 (i) と (ii) の課題を解決した．

本論文ではまず第 1 章において本研究の背景について述べ，第 2 章で関連する研究と本論文の位置づけを示す．

続いて第 3 章では音声対話システムにおいて (i) の課題に取り組む．この章では，デー

タベースを検索するタスクに関する音声対話システムにおいて，データベースの内容に依存せず頑健に動作する言語理解手法を開発した．ここではデータベースを検索するタスクで一般的に成立する対話文脈をモデル化して言語理解部に取り込み，音声認識誤りがある状況でも高精度に動作する音声言語理解手法を提案した．実際の対話システムを用いた評価実験により，提案法はデータベースを入れ替えた場合でも高精度に動作することが確認された．これにより，新規のデータベースに対しても頑健に音声言語理解を行うことが可能となる．

さらに第 4 章では音声対話システムにおいて (ii) の課題に取り組む．ここでは特定のドメイン知識に対応するサブシステムの集合によって構成され，新規のドメイン知識を容易に追加可能な音声対話システムを実現する．この枠組みにおいては，ユーザの発話がどのドメイン知識と関わるものかを判定するドメイン選択処理が重要である．本章では (ii) に対応するためドメイン選択問題を「複数のドメインから１つを選ぶ問題」ではなく,「現在話題としているドメインを継続するか否かを判定する問題」と捉え直したモデルを提案した．このモデルのもとでドメイン選択処理を構築することにより，新規のドメイン知識を自由に拡張可能な音声対話システムを実現した．

第 5 章では音声文書検索技術において (i) の課題に取り組む．新語や固有名詞などは検索語として重要であるにも関わらず音声認識の辞書から漏れることが多く，従来の音声文書検索技術では検出が困難であった．この問題に対し本章では，サブワードに基づく複数種類の検索手法を組み合わせることで高速かつ高精度な任意語彙の検索語検出技術を開発した．評価実験により，提案法は特に辞書外単語の検出において従来法より高速かつ大幅に高精度であることが確認された．これにより未知の検索語であっても頑健に動作する音声文書検索システムが実現される．

さらに第 6 章では音声文書検索技術において (ii) の課題に取り組む．音声文書検索において複数のシステムを自由に統合し，より高い精度で動作する音声文書検索システムを実現する．これまでも音声文書検索において複数のシステムを統合することにより検索精度が向上することが知られていたが，同時にインデックスサイズの増大を招くという問題が存在した．インデックスサイズの増大は，ストレージにかかるコストと検出速度を同時に劣化させる要因となる．本章では，未知語領域推定技術によって得られた未知語らしさをもとにして複数のインデックスを選択的に統合する手法を提案する．評価実験により，提案法はインデックスの統合による検索精度の向上を得ながらインデックスサイズの増大を抑えられることが確認された．

第 7 章では音声言語処理技術における本論文の貢献について述べる．また，本論文では扱いきれなかった課題や今後の方向性についても述べ，本論文を結ぶ．

Acknowledgements

This work was accomplished at Okuno & Yoshii Laboratory, Graduate School of Informatics, Kyoto University. During the program, I received numerous supports from a lot of wonderful people. I would like to express my profound acknowledgements to everyone.

First of all, I would like to express my sincere gratitude to my Ph.D. adviser, Professor Hiroshi G. Okuno. I have learned the fundamental attitude to address a research under his enthusiastic mentorship with numerous and highly suggestive comments. He gave me powerful encouragements to publish my own ideas to the world, and those experiences formed the basis of who I am today.

I also deeply thank my Ph.D. committee, Professor Tatsuya Kawahara, Professor Naofumi Takagi and Associate Professor Kazuyoshi Yoshii. They gave me a lot of valuable comments and suggestions to improve on the quality of this thesis.

All of my researches on spoken dialogue systems were conducted under the mentorship of Associate Professor Kazunori Komatani. He not only gave me professional comments for my research but also tutored me on the methodology of research. I would like to express my hearty thanks to him.

My research on multi-domain spoken dialogue systems was conducted in collaboration with Honda Research Institute (HRI). Dr. Mikio Nakano gave me the opportunity to study multi-domain spoken dialogue systems. Other members in HRI, Dr. Yuji Hasegawa, Dr. Kotaro Funakoshi, Dr. Johane Takeuchi, Dr. Toyotaka Torii, Dr.Kazuhiro Nakadai and Dr. Hiroshi Tsujino gave me a lot of valuable comments for my research. I would like to express my sincere gratitude to them.

I am also grateful to the past and present members of Okuno Laboratory, especially Professor Tetuya Ogata, Assistant Professor Katsutoshi Itoyama, Dr. Tsuyoshi Tasaki and Dr. Takuya Yoshioka. It was really great discussing with them sometimes until the early hours of the morning.

I have been working in Central Research Laboratory (CRL) in Hitachi Ltd. since

2006 and most of the researches on spoken document retrieval systems were conducted there. I would like to express my deep gratitude to my colleagues, especially Dr. Yoshinori Kitahara, Mr. Hisashi Ikeda, Mr. Nobuo Nukaga and Dr. Yasunari Obuchi. Dr. Yoshinori Kitahara, my first supervisor in Hitachi, supported me strongly to go for this Ph.D. course while continuing working in Hitachi. My current supervisor, Mr. Hisashi Ikeda and Mr. Nobuo Nukaga also gave me a lot of supports by willingly and heartily allowing me to go for the Ph.D. course. Dr. Yasunari Obuchi was not only my mentor during the training period in Hitachi but also a coauthor of most of my work in Hitachi. I am also grateful to my colleagues, Mr. Akio Amano, Dr. Hiroaki Kokubo, Dr. Hirohiko Sagawa, Dr. Masahito Togami, Mr. Takashi Sumiyoshi, Dr. Takeshi Honma, Mr. Yusuke Fujita, Mr. Yohei Kawaguchi, Dr. Qinghua Sun and Dr. Ryu Takeda. I really enjoy the daily discussions with you all. I also thank all managers of CRL who not only allowed me to go for the Ph.D. course but also supported me financially during the program. I am really proud of working in CRL with these great colleagues.

I would like to express my wholehearted gratitude to my parents for their continuous and infinite supports since my birth. Everything I have today is owed to them. I am also grateful to my grand parents who have given me incredible supports all through my life.

Lastly, and most of all, I would like to express my deepest gratitude to my wife, Akiko. She has given me unflinching support coupled with her incredible patience. Without her encouragement and endurance, this thesis would not have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Spoken language is one of the most natural communication tools; therefore, technologies for handling spoken language by using computers have been attracting much attention. For example, many systems with a speech interface, such as mobile phones or car navigation systems, have been investigated with the expectation that everyone can use them without a learning effort. For another example, speech mining systems, such as a reputation mining system from call center recordings, have also been attracting attention because vast amounts of daily recordings are gold mines of information. With advances in speech recognition techniques, some systems have reached to level of practicality, and there is a growing demand for applying spoken language technologies anywhere for any purpose.

Since spoken language is so familiar in our daily lives, users often expect spoken language systems to understand anything the users can. Imagine, for example, a tourist information system with a speech interface that can inform users about scenic sites. Users who want to make travel plans would ask the system not only about scenic sites but also about restaurants, hotels, transportation, weather, etc around these sites. Also imagine a system that can find specific recordings in call centers by the inputting of keywords. Such recordings contain many topics, sometimes including those about newly created products. Users would naturally expect the system to be able to search for recordings by inputting the name of products without considering whether the system has information about the products.

Unfortunately, most spoken language systems are designed for just limited domains, and they easily fail to understand what users said or the meaning of the recordings when

the speech is outside those domains. For example, while flight information systems [1] or bus information systems [2, 3] with a speech interface have been successfully developed, utterances those systems can understand are strictly limited to their domains. As a result, users need to speak carefully and know what utterances the system can understand. This severely degrades their ease of use. In another example, while news search systems, which can immediately retrieve specific recordings related to user queries, have been successfully developed [4, 5], the same systems cannot be applied to call center recordings because of the difference in vocabulary and language properties. Users of these systems must know which system should be used for what types of recordings.

As mentioned above, there is a large gap between what current systems can understand and what users expect. We addressed this gap by developing spoken language technologies that can handle a broad range of utterances by freely extending their knowledge resources. We call such technologies **open-ended spoken language technologies**. We believe this is a crucial step for spoken language technologies to be used by anyone for any purpose.

1.2 Target Systems

Our goal is to develop open-ended spoken language technologies. Note that detailed requirements for system architectures may be different for each system. In addition, especially for spoken language systems, speech recognition errors are inevitable and need to be considered in each layer of a system. Therefore, for addressing this issue, we investigated two target systems, spoken dialogue and spoken document retrieval, corresponding to two major aspects of speech; a man-machine interface and an information source. Constructing these two systems enables us to discuss open-ended problems from general perspectives.

Speech as Man-Machine Interface: Spoken Dialogue Systems Speech is a natural and powerful communication tool for humans. Therefore, it has the potential to be one of the most natural man-machine interfaces, not only due to the specific advantages of a hands-free or space-free interface (it can be used without large input device like a keyboard) but also by the easiness of using speech. A representative system for a speech interface is **the spoken dialogue system**, which is a computer system that can understand and respond to a user's spoken requests. An important feature of spoken dialogue systems is the ability to resolve ambiguity of meanings in

spoken utterances by using dialogue contexts, and sometimes by actively confirming this to the users. Such ability is useful for a speech interface because an utterance in spoken language is often shorter and more ambiguous than a written one.

Speech as Information Source: Spoken Document Retrieval Systems Since people use spoken language every day, vast amounts of speeches are produced. From this perspective, speech data can be seen as gold mines of information. For example, recordings in a call center contain information such as on the reputation of each product and problems with certain products. A representative system that treats speech as an information source is **the spoken document retrieval system**, which can find specific recordings from input queries. In this thesis, we mainly focus on a spoken term detection (STD) system, which finds positions of keywords, as a basis of spoken document retrieval systems. For spoken document retrieval systems, search accuracy and search speed are important metrics for measuring system efficiency, and a speech indexing module to achieve fast and accurate search is the main focus of this research. In addition, index size and indexing time have to be considered in terms of server cost.

1.3 Centralized vs. Distributed Architecture

There are two major architectures for developing a system that has multiple-domain knowledge. One architecture is a centralized architecture, in which one large module manages many types of domain knowledge. The centralized architecture can incorporate any knowledge in a direct manner no matter how complex the knowledge is. However, updating such a system becomes increasingly difficult according to the increase in knowledge because its developers need to consider all the effects of modification.

The other architecture is distributed architecture, in which many subsystems on different domains work in a coordinated manner (e.g., distributed spoken dialogue systems [6–13]). While an integration method of multiple domain systems are necessary for achieving consistent system operation, each domain system may be developed without taking into account other domain systems. This makes the distributed architecture easy to maintain even when the domain knowledge becomes large.

While the distributed architecture seems to be promising for developing open-ended systems, many problems remain unsolved. First, no matter how much knowledge is in-

corporated into a system, there could be utterances that the system is not designed for. Second, a good integration method of multiple domain systems is essential. In particular, if the integration method is heavily dependent on the domain knowledge in the system, updating the system becomes difficult. In addition to the above problems, especially for spoken language systems, speech recognition errors need to be considered in all layers of the system, which makes the system more complicated. In this thesis, we basically use the distributed architecture and investigate those problems towards open-ended spoken language technologies.

1.4 Technical Problems and Solutions

Based on the above arguments, the technical problems we targeted are summarized as follows.

Problem 1: Spoken language analysis with little or no domain knowledge.

No matter how much knowledge the system has, there can be an utterance on which the system has little or sometimes no information. Therefore, it is necessary for open-ended systems to analyze such utterances without heavily depending on specific-domain knowledge. For spoken language systems, speech recognition errors are inevitable. Therefore, robustness against speech recognition errors must be complemented in a domain-independent manner.

Problem 2: Integration of knowledge on arbitrary number of domains.

If the integration method of multiple domain knowledge is heavily dependent on the domain knowledge in a system, the system becomes increasingly difficult to update according to the increase in domains. In addition, the system normally becomes costly according to the incorporation of new knowledge; therefore, integration efficiency should also be pursued. Finally, speech recognition errors must be considered when integrating spoken language systems.

We have been studying the above problems for several years, and this thesis reports on the results of these studies. Our solutions for these problems are summarized as follows;

Solution 1: General-domain spoken language analysis based on domain-independent word/context models.

To robustly interpret utterances without depending on specific-domain knowledge, we focus on domain-independent modeling of vocabulary and dialogue context.

- We first propose general context models for dialogues on database search tasks. Different from conventional studies, the proposed context models are domain-independent; therefore, they can be used even when replacing the background database. We also propose a spoken language-understanding method based on these context models, and this method is highly accurate and robust against speech recognition errors. More importantly, it exhibits almost the same accuracy even when replacing the background database.
- We also propose a vocabulary-independent spoken term detection system, which can detect positions of arbitrary keywords from a speech database. We also tandemly combine three subword-based systems, and narrow the search space in a stepwise fashion. Experimental study indicates that the proposed system is fast and much more accurate than conventional systems, especially when searching for out-of-vocabulary keywords.

Solution 2: Domain-extensible integration of multiple systems based on robust domain selection method.

For developing domain-extensible systems, we investigated selective combination methods of multiple systems.

- We first propose a domain selection method for developing domain-extensible spoken dialogue systems. The important feature of this method is that it can handle even newly created subsystem that handles new domain knowledge (we call the subsystem "domain experts"). Developers can create domain experts independently with this method. An experimental study indicates that our domain selection method is more accurate than conventional methods.
- We then propose a selective index combination method for developing a compact and accurate spoken term detection system. Many index combination methods have been proposed. However, they have a defect in that the index size becomes large according to the increase in combinations. The increase in index size leads to both slow search speed and high storage cost; therefore, we focus on an efficient index combination method that would not result in

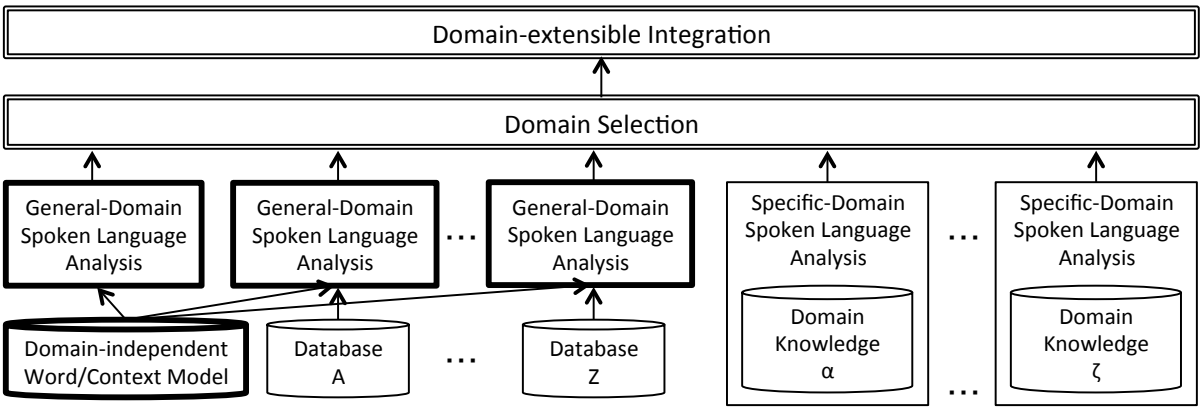


Figure 1.1: Basic architecture towards open-ended spoken language technologies.

an increase in index size. We selectively combine only valuable indices and use new selection criteria based on an out-of-vocabulary region estimator. The proposed method achieves high search accuracy by suppressing the increase in index size.

The basic architecture we used is illustrated in Figure 1.1. In this architecture, many spoken language analysis modules work in parallel then the results from those spoken language analysis modules are selected and integrated. Modules outlined in bold lines and double line correspond to solutions 1 and 2, respectively. We developed both spoken dialogue and spoken document retrieval systems based on this architecture. The following chapters describe the details of these studies and discuss the essence of our open-ended spoken language technologies.

1.5 Organization of the Thesis

The organization of this thesis is shown in Figure 1.2.

In Chapter 2, we review the literature on conventional spoken dialogue and spoken document retrieval systems. We also discuss open-ended systems in other research areas.

In Chapter 3, we describe our robust language-understanding method for dialogue systems on the database search task. This method can be used even when the database is not the one that the system is prepared for. We also propose general context models for the database search task and incorporate context information from the models into the language understanding procedure.

In Chapter 4, we describe our spoken dialogue system that consists of multiple subsystems on different domains and that can easily extend new domain knowledge. The key component of this system is a domain selection method, which select an appropriate domain for each user utterance. We proposed a new domain selection scheme to determine whether the previously-domain should be kept or not. Experimental results indicate that the proposed domain selection method achieved better results than conventional methods with achieving the ability to extend new domain knowledge.

In Chapter 5, we describe our spoken document retrieval system with new indexing and searching methods that can detect positions of arbitrary keywords from a speech database. To achieve fast and accurate term detection, we tandemly combined three types of open vocabulary indexing methods. Experimental results indicated that the proposed method clearly outperformed conventional spoken term detection methods, especially when searching for out-of-vocabulary keywords.

In Chapter 6, we describe our index combination method for STD systems. Simply combining many indices increases index size, which raises the problems of high storage cost and slow search speed. To suppress the increase in index size, we use a selective index method based on an out-of-vocabulary region classifier. The proposed method is confirmed to be able to suppress the increase in index size while improving search accuracy.

Finally, in Chapter 7, we conclude the thesis with a discussion on the contributions of our study and future directions.

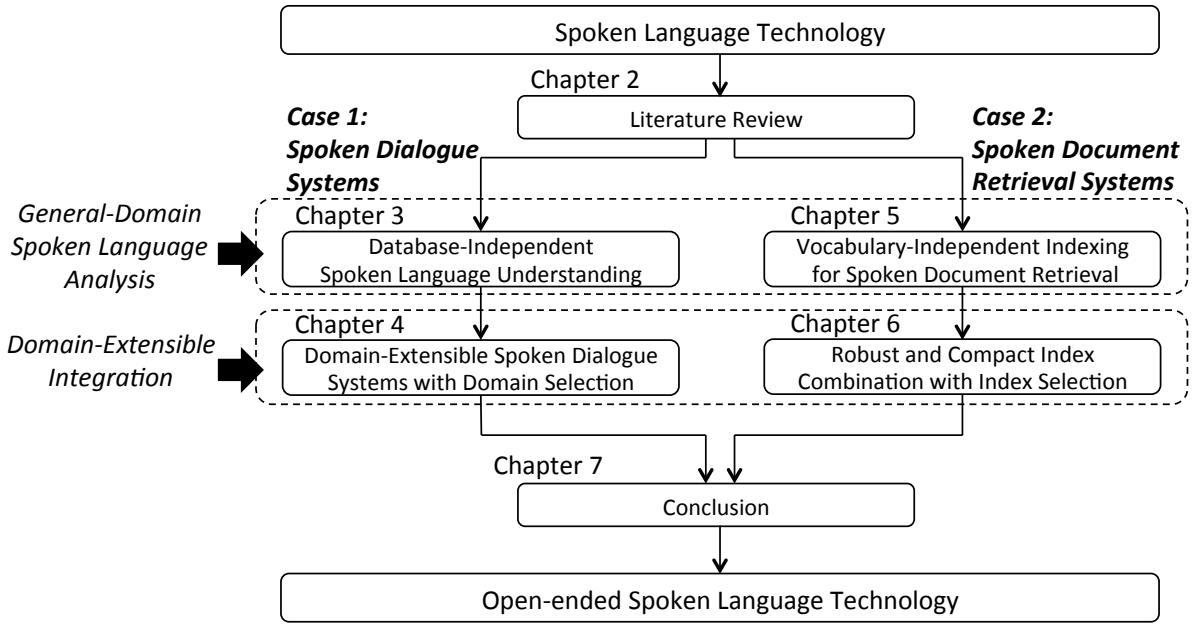


Figure 1.2: Organization of the thesis.

Chapter 2

Literature Review

2.1 Spoken Dialogue Systems

2.1.1 Dialogue Systems based on Deep Domain Knowledge

A spoken dialogue system is a computer system that can understand a user 's utterance and output an appropriate response. A spoken dialogue system often consists of speech recognition, natural language understanding, dialogue management, natural language generation, and speech synthesis modules. Since users often speak when a system is speaking (known as barge-in), some systems have architecture in which each module works in parallel [14, 15].

One of the root systems of spoken dialogue is SHRDLU [16], which is a computer system that can interpret user directions based on deep semantics of a building block world. Although SHRDLU can understand a very limited number of utterances related to a building block world and cannot understand spoken language, it has inspired many subsequent dialogue systems including expansion for spoken language. In the 1990's, spoken dialogue systems, such as VOYAGER [17] and ATIS Project [1], were developed, which created the basis of current spoken dialogue systems. In the the first decade of the 2000s, many systems were put into practice, for example the bus guide systems in Kyoto [2] or Pittsburgh [3]. Recently, many studies have been conducted to introduce machine learning theory to improve language understanding [18] and dialogue management modules [19].

These systems are based on deep domain knowledge for precise understanding of user utterances. To incorporate dialogue context, these systems have a kind of internal state such as finite state automaton or information state. Such systems, however, often exhibit a lack of robustness for out-of-domain utterances due to the difficulty in creating deep

domain knowledge for arbitrary domains.

2.1.2 Question-Answering-based Systems

Recently, spoken dialogue systems based on question-answering systems have been developed and become main stream commercial dialogue systems. The root of these systems is ELIZA [20], which responds to user utterances based on simple word matching. Although ELIZA does not have any deep domain knowledge, it can respond to a variety of user utterances. In the first decade of the 2000s, many systems were developed based on the same idea to cope with a variety of user utterances. One of the earliest studies on spoken dialogue systems was conducted by Takemaru [21]. This system can answer user questions based on a vast amount of question-answer pairs. Recently, similar agent systems for mobile phones were developed and put into commercial use, for example Siri [22] and Shabette Concier [23].

Although these systems can respond to various types of utterances, they basically have little or no internal states. As a result, they cannot understand dialogue context, which means they inherently lack the ability to actively talk with users; they only answer questions. In addition, they do not have the ability to understand ambiguous or erroneous utterances with speech recognition errors by referring to the dialogue context, which is frequently observed in human-human conversation.

As described above, question-answering-based systems can cope with a broad range of utterances but lack the ability of robust understanding. In Chapter 3, we tackle this problem and propose a spoken language-understanding method, which is much more robust, and at the same time, can be used for any database search task.

2.1.3 Combination of Multiple Systems

There have been studies on spoken dialogue systems that consist of many subsystems on different domains. For example, [24] proposed a system in which each subsystem has a speech recognizer and the subsystem with most probable speech recognition results responds to the user. A bias for a previously selected subsystem has been introduced [6,8]. Those methods sometimes work; however, their domain selection accuracy is not sufficient because they do not fully use the dialogue history. Furthermore, there is a hidden problem in that these methods cannot detect a problematic situation in which neither the most probable domain nor the previously selected domain is incorrect.

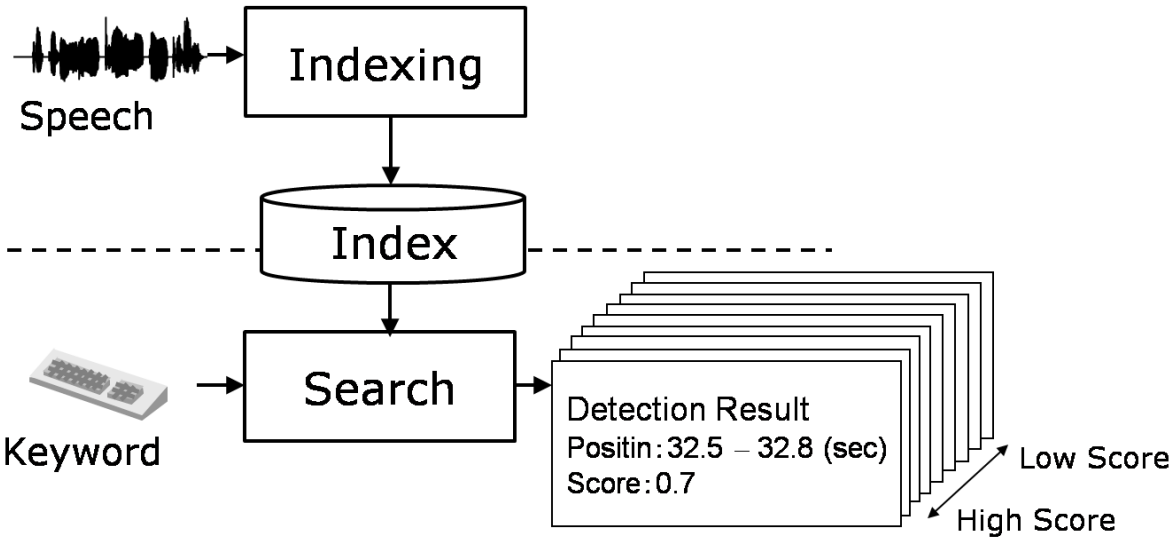


Figure 2.1: Overview of spoken term detection system.

In Chapter 4, we tackle these problems in more detail and propose an accurate and extensible domain selection method.

2.2 Spoken Document Retrieval Systems

With recent progress in storage devices and networks, a vast amount of multimedia data has been published on the Internet and accumulated in personal computers. Vast amounts of speech and video data have also accumulated in call centers or in the broadcasting industry for their use. To make the best use of these speech data, an efficient search method is necessary. There was an attempt to use manually created metadata. However, creating such metadata is expensive and it is difficult to create metadata that fully cover user interests. Therefore, there is an expectation in developing a spoken document retrieval system [5], which is a computer system that can interpret the content of multimedia data and detect the content from user directions.

In this thesis, we focus on STD systems, which can immediately detect the positions of keyword uttered in hundreds of hours of speeches. Spoken term detection is key for spoken document retrieval, and many studies have been conducted, including those in the NIST STD [25] and NTCIR SDR [26, 27] workshops.

In general, an STD system consists of indexing and search modules, as described in Figure 2.1. The indexing module converts a speech wave into an index for efficient keyword searching. Normally, the indexing module works only once when new speech

is added to the system. The search module works when the user inputs a keyword and detects the positions of the keyword uttered in the speeches referring to the index. The search module often outputs not only the positions of the keyword but also its detection score, which indicates the plausibility of the detection results.

2.2.1 Word-based Indexing of Spoken Document

Conventional methods related to STD can be classified into two types; word-based and subword-based.

In the indexing module of word-based methods, a large vocabulary continuous speech recognition (LVCSR) system is first used to convert a speech wave into word sequences. When searching a keyword, word matching is conducted for detecting keyword positions. An index structure, such as an inverted index, was usually used for fast keyword searching [28]. It is also common to use word lattices or confusion networks for improving recall of detection results [29,30]. Such methods were confirmed to be fast and accurate for speech data for which the language modeling is relatively easy such as news speech. However, there is a problem in that if the domain of speech data is very different from that of language models, search accuracy severely degrades. Furthermore, if the keyword is not contained in the vocabulary of the LVCSR system, such a keyword cannot be detected.

2.2.2 Subword-based Indexing of Spoken Document

To overcome the vocabulary limitation of word-based indexing methods, subword-based indexing techniques [31–35] have been widely used. A subword is a morphological unit that is smaller than a word, for example phonemes or syllables. In the indexing module of subword-based methods, subword recognition is first used to convert a speech wave into subword sequences. In the search module, the keyword is converted into a subword sequence by using certain rules. Then, keyword positions are detected by calculating the distance between the recognized subword sequence and the subword expression of the keyword. Many researchers have used edit distance or its modification for measuring distance [36–38]. A method based on subword recognition has the advantage that it ideally can detect arbitrary keywords. It is also beneficial that the time needed for indexing is relatively small because subword recognition has normally a smaller hypothesis space compared to the LVCSR system. On the contrary, a subword-based method is normally less accurate compared to a well trained LVCSR-based method [29] because a

subword-based method often detects false positives that have similar pronunciation. It is also known that the search speed is slow due to the complex procedure to calculate the distances between subword sequences.

There have been several studies on making subword-based method fast. Katsurada et al. [39] proposed a fast search method based on a suffix array. Kanda et al. [40] and Yu et al. [41] proposed using an inverted index of the phoneme N-gram. Nakagawa et al. [42] proposed using a suffix array based on the syllable tri-gram. These methods enable fast search compared to the simple edit distance search. However, they are an approximation of the complex edit distance calculation, and search accuracy is normally not higher than the edit distance search.

For improving the search accuracy of subword-based methods, Itoh et al. [37] and Dharanipragada and Roukos [43] proposed using a word spotting method after subword-based methods for accurately re-ranking the detection results. However, there is a problem in that the search speed are very slow; Dharanipragada and Roukos ' method required 15 seconds to search for a keyword from 10-hour speech data [43], and Itoh et al. 's method required over 1 second to rescore one search candidate [37].

As mentioned above, conventional open vocabulary systems are either slow, inaccurate, or both. In Chapter 5, we tackle this problem and propose a subword-based spoken term detection system that is faster and significantly more accurate than conventional systems.

2.2.3 Combination of Multiple Indices and Its Defect

There have been studies on combining an LVCSR-based method and a subword-based method. For example, some studies involved using a LVCSR-based method for keywords included in a LVCSR system's vocabulary and a subword-based method for unknown keywords [29, 42, 44]. However, there is still a problem with subword-based methods, i.e., low accuracy and slow search speed, when searching unknown keywords.

Recently, methods that combine multiple types of indices have been proposed, which achieve high detection accuracy [45–47]. For example, the best performance in the latest NTCIR STD evaluation [26] was obtained using a method that combines ten different recognizers' outputs [47]. There are many variations in index-combination methods: subword unit type (word/phoneme [32,33], original subwords [48]), index format (lattice [32], confusion network [33,46,47]), score calculation (modified edit-distance [47], and weighted-sum [46]).

A defect with the multiple index-combination method is its large index size. As many indices are combined, the index size becomes larger. A larger index not only increases storage cost but also slows search speed [26,47]. A confidence measure (like the one in [49]) could be used to identify the redundant portions of an index made from a single recognizer [50–52]. However, it is not always easy to extend this method to a combined index made from multiple recognizers because confidence measures from different recognizers are often biased differently. Furthermore, a confidence measure of a region that contains Out-of-Vocabulary terms (OOVs) tends to have a small value; therefore, confidence-measure-based index pruning may degrade the accuracy for OOV queries.

In Chapter 6, we address the above defect and propose an index combination method that can suppress the increase in index size while improving accuracy.

2.3 Position of the Thesis

Figure 2.2 shows the position of this thesis. We plotted the above studies from the perspectives of the target use (horizontal axis) of the systems and limitation in domain knowledge (vertical axis). Some representative systems for written language are also shown as a reference. We classified each system into three categories based on its ability to handle domain knowledge; specific-domain, general-domain, and open-ended.

Specific-domain systems can handle only limited utterances related to specific domains. They often show a lack of robustness against out-of-domain utterances. Despite this limitation, they normally can deeply interpret natural language by referring to dialogue context or by using specific word knowledge. Dialogue systems based on deep domain knowledge [1–3, 17, 20] or spoken document retrieval systems with word-based indexing [4, 28] are classified as specific-domain systems.

General-domain systems rely on little or no domain knowledge; as a result, they can handle a wide range of utterances. A basic question-answering-based dialogue system [21] and spoken document retrieval systems with subword-based indexing [53,54] are classified. We also classify commercial systems based on question answering systems [22, 23] as general-domain systems, even though they sometimes rely on hand-crafted ontology [22]. These system can cope with various types of utterances; however, they often lack the ability to deeply interpret utterances and robustness against speech recognition errors.

Open-ended systems can handle a wide range of utterances, and at the same time,

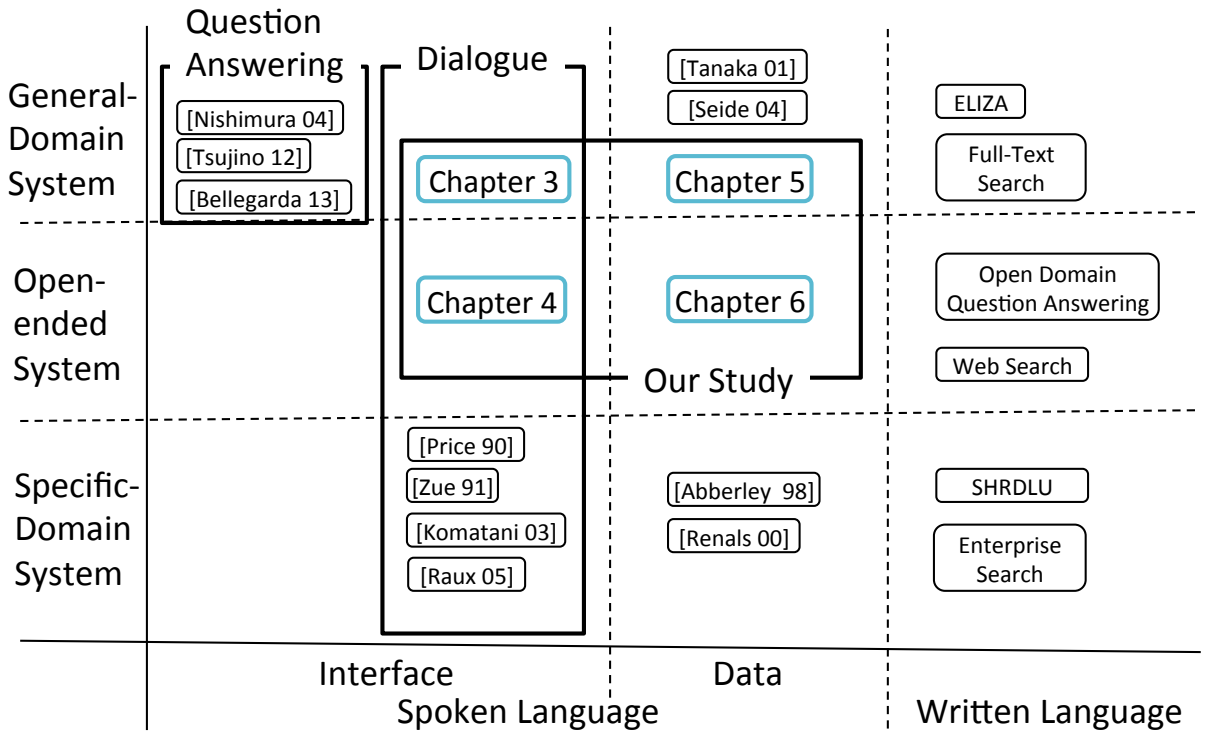


Figure 2.2: Position of the thesis.

can extend their knowledge freely, i.e., they need to handle as wide a range of utterances as with general domain systems and extend their knowledge as deeply as with specific domain systems. Open domain question-answering systems [55–57] for written language can be categorized in this category. Current web search systems often produce general search systems with multiple specific search systems (e.g., for news, scholarly papers, blogs, etc.), and we categorize these systems as open-ended systems.

Chapter 3

Database-Independent Spoken Language Understanding for Database Search Task

3.1 Introduction

In this chapter, we focus on a language understanding module for dialogue systems in database search tasks such as a restaurant search system or a hotel search system. Because speech recognition errors are inevitable, the language understanding module needs to robustly extract user's intention without being misled by speech recognition errors. Although there have been many studies on rejecting speech recognition errors by using confidence measures [58] or acoustic likelihood [59, 60], there are few studies that utilize dialogue context for robust language understanding. In this study, we developed a new language understanding method that incorporates context information in the database search task.

In addition to the difficulties posed by speech recognition errors, a spoken utterance normally consists of fewer words than a written sentence, which makes the utterance more ambiguous to understand. For example, in a typical restaurant search system, the simple utterance " credit card " can have two meanings: " I want to search for a restaurant where I can pay by credit card " or " I want to know which types of credit card I can use at the restaurant ". A language understanding module for the database search task should be able to resolve such ambiguity by referencing context information. In this study, we propose resolving such ambiguity by using a decision tree classifier trained on a dialogue corpus.

There have been a few studies on training dialogue systems with dialogue corpora with

hand-crafted labels [61–64]. However, collecting individual items of dialogue in a corpus and labeling them is normally quite costly, and it is desirable that a trained module be usable in other systems. We call this ability “ domain exchangeability ”. In this study, we first proposed context models that represent general dialogue contexts for a database search task and then trained a language understanding module using features derived from these models. This gave our language understanding module the ability of domain exchangeability, which was confirmed by experiments on a restaurant search system and a hotel search system.

The structure of this chapter is as follows. In Section 3.2, we review previous studies related to the proposed language understanding method. Next, we present the context models we developed for representing dialogues in a database search task in Section 3.3 and explain the language understanding method using these models in Section 3.4. We describe the results of experimental evaluation in Section 3.5 and then conclude this chapter in Section 3.6.

3.2 Related Studies

In this chapter, we describe a spoken language understanding method that incorporates dialogue contexts. One related work [65] describes using hand-crafted rules that incorporate dialogue contexts for spoken language understanding. However, it is expensive to make hand-crafted rules because of the significant effort required of assorted experts. Furthermore, such rules are normally domain-dependent, which means a developer has to create different rules for different dialogue systems even if the two systems have a very similar structure.

For incorporating dialogue contexts, [61] and [62] utilized dialogue-act probability estimated from dialogue corpora that contain dialogue-act labels. They modeled the probability using an n-gram model. In their work, fine-grained and domain-dependent representation of dialogue acts, such as “SET-START-TIME-OF-MEETING-ROOM” or “REQUEST-FOR-PARKING-SPACE”, are used to obtain meaningful information from estimated dialogue-act probability. However, with these methods the dialogue corpora need to be obtained and labeled before constructing the dialogue system, which is normally a costly task. Moreover, when applying such representation for dialogues in database search tasks, the number of dialogue acts increases in accordance with the number of

items in the database and eventually an unrealistically large dialogue corpus to train the dialogue-act probability is necessary, especially when there are many items in the target database.

[66] proposed using dialogue-related features such as the utterance type of a previous system response when constructing a classifier that decides whether to accept or reject a language understanding result. In a similar way, [63] and [64] utilized dialogue features to decide on the acceptance or rejection of utterances. They reported improvements to the accuracy of the acceptance/rejection decision-making process in their target domain, but they did not consider extending their models to other domains.

In our method, instead of modeling the fine-grained and domain-dependent representations of contexts, domain-independent representations of contexts suited for general dialogues in database search tasks are proposed and utilized. We trained a decision classifier that references features obtained from our context models and used it to robustly understand user requests from speech recognition results. We also propose for the first time two dialogue models for the database search task: a dialogue progress model and a history structure model, both of which are introduced to capture general but fine-grained contexts.

We should point out that the features obtained from our context models are dependent on the database search task but independent of database domains. Therefore, once a language understanding module has been developed, it can be used for many different databases without having to change anything.

3.3 Context Model of Database Search Task

3.3.1 Dialogues in Database Search Task

As stated above, we newly propose two context models for the database search task. Before describing these models, we explain the characteristics of the database search task in more detail.

In our method, we assume a database search task as a task that accesses a relational database to obtain information from the database. A relational database is a specific type of database in which each entry consists of multiple attributes and their accompanying values. We assume the database has at least one key attribute that can uniquely identify an entry. An example of such a database is shown in Table 3.1. In the following expla-

Table 3.1: Example of relational database (restaurant domain).

Attribute	Value
Restaurant Name (key)	Cafeteria Kusunoki
Food Type	Japanese
Explanation	An inexpensive cafeteria chain that is common in Kyoto ...
Address	Yoshida, Sakyo-ku, Kyoto city
Telephone	555-5555
Opening Hours	18:30-23:30
Holiday	Thursday
Access	A 10-minute walk from Demachi-yanagi station
Credit Card	JCB, VISA
Parking	2
Budget Min	400 yen
Budget Max	1,000 yen

nation we assume a restaurant database, but any type of relational database can be used in our model.

An example dialogue in the database search task is shown in Figure 3.1. The database search task differs from the slot filling task [67] in two ways:

- The information slots required for task completion are different for each user as opposed to the slot filling task, where required slots are defined beforehand. For example, while some users might want an inexpensive restaurant, others are more interested in the nearest restaurant.

- User goals might vary during a dialogue. For example, users might need to change their search conditions after checking the search results if there is no restaurant that sufficiently matches their intention. They also might change their mind if they find a good restaurant that they were not expecting.

If all statuses in a dialogue are well represented, they can function as effective constraints for a language understanding module that robustly parses speech recognition results. For dialogue systems that manage slot filling tasks (weather information, bus information, etc.), the flow of dialogue can be explicitly represented as a finite-state automaton [68]. Therefore, it is natural to use the status of the automaton for robust language understanding. However, as mentioned previously, the dialogue flow in a database search task cannot be represented beforehand, so an additional representation of dialogue flow is needed to express dialogue status.

S1: This is a restaurant guide system. What kind of restaurant would you like?

U1: **I'd like to find a restaurant in Gion.**

S2: I found 259 restaurants in Gion.

U2: **Are there any Japanese restaurants?**

S3: I found 51 Japanese restaurants in Gion.

U3: **Maximum 3000 yen.**

S4: I found 51 Japanese restaurants in Gion, maximum 3000 yen.

U4: **How about setting the maximum to 1000 yen?**

S5: I found 2 Japanese restaurants in Gion, maximum 1000 yen. Yoshida Restaurant and Cafeteria Kusunoki.

U5: **Where is Cafeteria Kusunoki?**

S6: The address of Cafeteria Kusunoki is Yoshida, Sakyo-ku, Kyoto city.

Figure 3.1: Example dialogue in database search task.

We mentioned in Section 3.2 that if the dialogue status (in other words, the context) is dependent on the domain, it becomes difficult to port the acquired knowledge to a newly created system. In our work, we take special care to make our dialogue status independent of the domain while at the same time ensuring that the status has sufficient granularity to realize robust language understanding.

In the next two subsections, we explain two proposed representations of dialogue flow in a database search task.

3.3.2 Dialogue Progress Model

We first define a dialogue progress model that satisfies the requirements above. This model is based on our observation that a typical dialogue in a database search task starts with a user's request (which includes various search conditions) to find desired entries and then, after the desired entries from the databases have been found, moves on to the user ' s request for specific attributes related to the entry. For example, in the restaurant search task, a typical user first inputs search conditions (value, place, etc.) to find a desired entry (restaurant) and then requests the specific attributes (address, telephone number, etc.) of the provided entry.

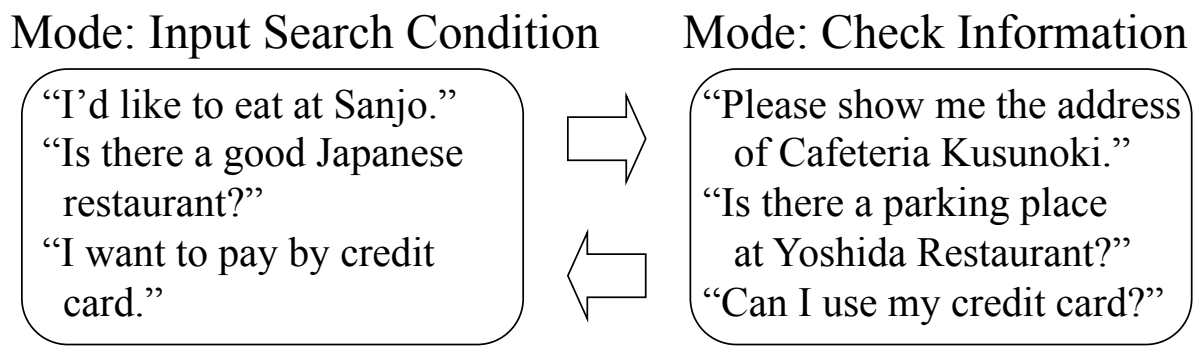


Figure 3.2: Two modes of the dialogue flow model.

We defined two modes for representing the two elements above: “input search condition” for the first one and “check information” for the second. Additionally, we assume that the dialogue in the database search tasks can be coarsely represented by the transition between these two modes (Figure 3.2). Note that if a user is not satisfied with the search results, he or she will change the search conditions, so it is possible to change the dialogue mode from “check information” to “input search condition” when necessary. We call this model the **“dialogue progress model”**.

Many features, such as the current mode or the mentioned attributes in each mode, can be derived from the dialogue progress model and used for our language understanding module. For example, in Figure 3.1, utterances U1—U4 were uttered under “input search condition” mode and U5 was uttered under “check information” mode. Such information can be used to robustly understand the user’s intention from speech recognition results that may contain errors.

3.3.3 History Structure Model

In the database search task, different users have different search demands. If a system is able to identify which search condition is most important to the user, it can prevent the important condition from being overwritten by speech recognition errors. To approach this idea, we set the following assumption:

- A search condition that is not changed for a long period is the most important.

To express this assumption, we propose representing the history of the search conditions (attribute and value pair) as a tree structure. The proposed tree structure is constructed so that each node in the tree represents one search condition, and a node (a

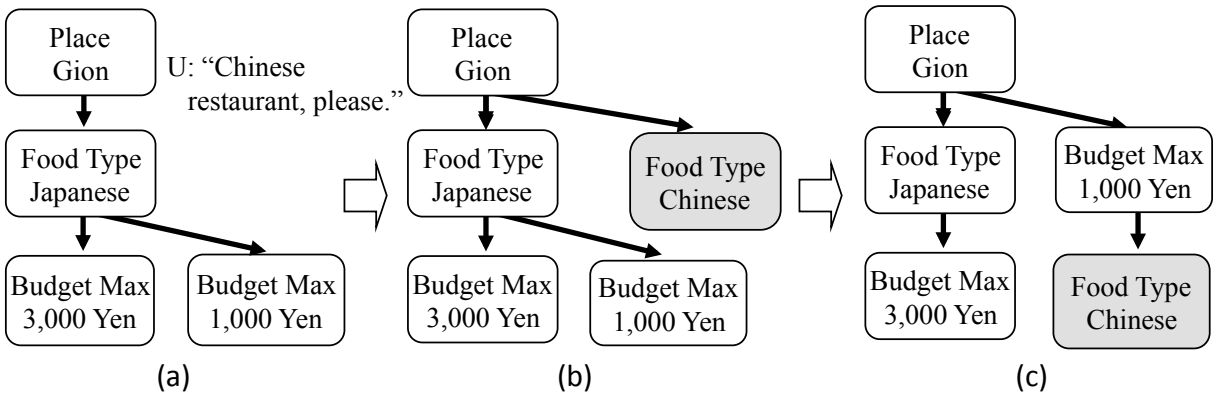


Figure 3.3: Example of tree-structured dialogue history.

search condition) that is not changed for a long period is designed to be on top of the tree and to have many children. We call this structure the **history structure model**. We manage the nodes of the tree according to the following rules:

1. New search condition is put on the bottom of the tree.

2. Current search conditions are put on the right-most side of the tree.

3. If the attribute of a new search condition is the same as the attribute of one of the current search conditions (we call this node A), a new brother node is added to the right of A and the right-most side of A's child is moved so as to satisfy rules 1 and 2 above.

Note that rule 3 is for managing updated search conditions. According to these rules, a node (a search condition) that is not changed for a long period moves to the upper part of the tree and has many children. By constructing this tree structure, we can extract features such as the position or the number of children of the search condition in which we are interested, focusing on those features that represent the most important search conditions for the users.

Figure 3.3 (a) shows an example of a tree constructed after a user input the search conditions "Place: Gion", "Food Type: Japanese", "Budget Max: 3,000 Yen", and "Budget Max: 1,000 Yen", in this order, for searching restaurants. Current search conditions are represented as the right-most path of the tree. If the user requests "Chinese restaurant, please", the system can understand the utterance and add a node that represents "Food Type: Chinese" to the right of "Food Type: Japanese", as in Figure 3.3 (b). Then, the

right-most side of the child of "Food Type: Japanese" is moved to the position between "Place: Gion" and "Food Type: Chinese" so as to satisfy rules 1 and 2 above. This results in the tree shown in Figure 3.3 (c). From this tree, we can conclude that the search condition "Place: Gion" is important to the user. Such information can be used to robustly reject unintentional updates of search conditions caused by speech recognition errors.

3.4 Spoken Language Understanding with Dialogue Contexts

In this section, we describe how to incorporate the dialogue contexts described in Section 3.3 into our spoken language understanding module. Figure 3.4 shows an overview of our spoken dialogue system for a database search task. Our system consists of a speech recognition module, a language understanding module, and a dialogue management module. User utterances are processed as follows.

1. The speech recognition module converts a user utterance into a word sequence.

2. The language understanding module estimates the dialogue act and content words by referencing the speech recognition result and the dialogue contexts. Specifically, the language understanding module first calculates the similarity between the speech recognition result and putative utterances prepared for each dialogue act. The content words are then extracted from the speech recognition results by dictionary matching. For each content word, the confidences of dialogue acts, including the possibility of rejection of the content word, are estimated using a decision classifier. Finally, the confidences estimated for each content word are integrated and the dialogue act and accepted content words for that utterance are decided.

3. The dialogue management module updates the dialogue states based on the output from the language understanding module and responds to the user accessing the database.

Note that calculating the similarity between speech recognition results and putative utterances corresponds to the technique in a previous study [70].

In this study, content words are defined as attributes and values that appear in the relational database. We defined two dialogue acts, "Addition of Search Conditions" and

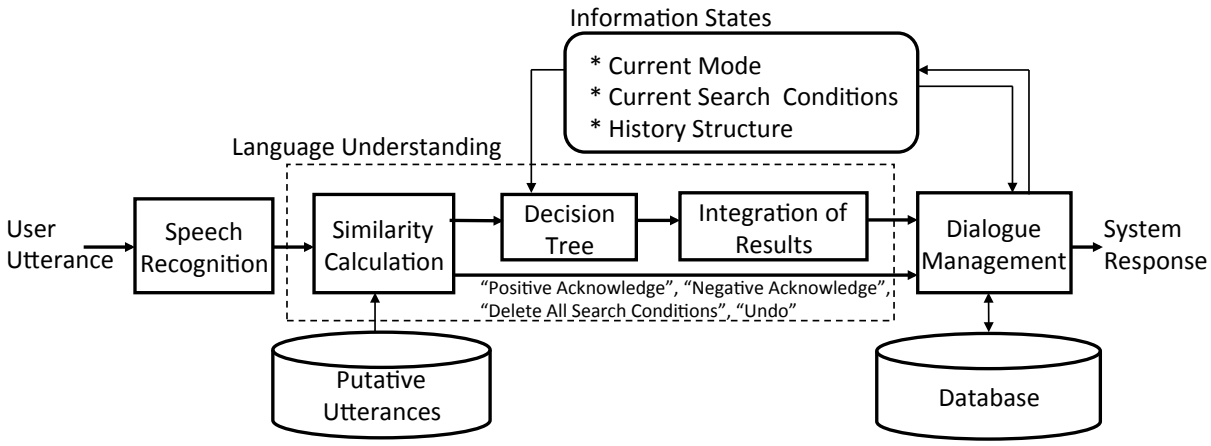


Figure 3.4: Overview of our system for database search task.

"Deletion of Search Conditions", for setting the search conditions, and one dialogue act, "Check Information", for accessing database entries. We also set four additional dialogue acts, "Positive Acknowledge", "Negative Acknowledge", "Delete All Search Conditions", and "Undo", for easy use of our database search system. In the next section, we describe the process flow of our language understanding module in more detail.

3.4.1 Similarity Calculation with Putative Utterances

In the language understanding module, first, similarity between the speech recognition results and the putative utterances prepared for each dialogue act are calculated. In this study, we follow the similarity calculation procedure outlined in [70][1]. The similarity of the most similar utterances prepared for a dialogue act indicates how likely the speech recognition result is to be that dialogue act[2]. An example of a putative utterance is "Please show me restaurants that serve good *FOODTYPE*", which corresponds to the dialogue act "Addition of Search Conditions". Here, *FOODTYPE* corresponds a "Food Type" attribute in the restaurant database. We prepared 509 such putative utterances for the restaurant domain.

After calculating the similarity, we apply the decision tree classifier for each content word and outputs from the classifier are then integrated into one interpretation of the

[1]We slightly modified the weight calculation for each content word. Specifically, we calculated the weight of each content word as a product of syntax weight and its recognition confidence [49].

[2]In this study, although our similarity technique was inspired by the one in [70], we do not rely on a single specific similarity measure. We feel that other similarity measures, such as similarity based on the vector space model [71], can also be used for our language understanding method.

S1: The most similar dialogue act in similarity calculation.

S2: Similarity of S1.

S3: Similarity of the second most similar dialogue act in similarity calculation.

S4: S2/S3.

S5: Type of the content word (attribute, value, key attribute, key value).

S6: Word confidence measures.

S7: Existence of an attribute that can be paired (e.g., "Budget Max" and "1,000 Yen").

Figure 3.5: Features obtained from a single utterance.

user's intention. However, note that the four dialogue acts, "Positive Acknowledge", "Negative Acknowledge", "Delete All Search Conditions", and "Undo", do not contain any content words. Therefore, if the similarity of one of these is higher than one in the other dialogue acts, our language understanding module simply defines this act as the dialogue act of the utterance.

3.4.2 Application of Decision Tree for Content Words

The next step is applying the decision tree for each content word. We apply the decision tree for each content word w_i based on a feature set F_i to calculate a confidence measure $CF(s|F_i, w_i)$[3] of a dialogue act s or a confidence measure $CF(s|F_i, w_i)$ that indicates the confidence of a rejection of w_i. Here, s satisfies $s \in \{$" Addition of Search Conditions ", " Deletion of Search Conditions ", " Check Information "$\}$.

We prepared 33 types of features F_i, each of which was derived from each content word. Features are classified into three types: **utterance-based features** (Figure 3.5), **general context features** (Figure 3.6), and **context features for database search task** (Figure 3.7).

As utterance-based features, we used, for example, the results of similarity calculation (S1 - S4) and word confidence measure (S6). Note that for S2 and S3, if there are multiple dialogue acts having highest similarity, an "ambiguous" label is used.

As general context features, we used, for example, a previously selected dialogue act

[3]In a decision tree, CF is calculated for each leaf as $(M + 1)/(N + P)$ [72], in which N indicates the number of samples associated with the leaf, M indicates the number of samples of s (or "reject") associated with the leaf, and P indicates the number of classes.

G1: Previously selected dialogue act.

G2: Whether the previous system utterance is a question.

G3: Whether the content word had been confirmed.

G4: Whether the content word had been denied.

G5: Whether the content word had been deleted.

Figure 3.6: General context features.

(G1) and whether the content word had been denied (G4). Context features for the database search task are original features derived from the proposed context models. We prepared many features derived from the dialogue progress model, for example, current mode (C1) or the number of entries that satisfy current search conditions and are mentioned after entering "Check Information" mode (C3). A feature C7 indicates the ratio between the number of entries matched for current search conditions and the number of such entries mentioned after entering "Check Information" mode. A feature C8 indicates the ratio between the number of entries matched for current search conditions and the number of such entries mentioned during the dialogue. We also prepared features derived from the history structure model, for example, depth of the node overwritten by the content word (C14) and the number of children of the node overwritten by the content word (C16). Note that the features C18, C19, C20, and C21 are depth-related features normalized by using the depth of the current tree structure.

We trained a decision tree by using dialogue data, where each content word in each recognition result is labeled with its corresponding dialogue act or a "rejection" label. Note that the "rejection" label is used if the content word was a speech recognition error. After training, the decision tree can output confidences for each dialogue act and a confidence of rejection, in parallel, based on the features described above.

3.4.3 Integration of Outputs from Decision Tree

Finally, the language understanding module integrates the outputs of the decision tree for each content word to decide a dialogue act and then selects the content words that the system accepts. Specifically, this entails the execution of the following procedure.

1. Select the dialogue act S of the utterance by using the confidence measure

C1: Current mode in dialogue progress model. (Initial Status: Addition of Search Conditions.)

C2: The number of entries that satisfy current search conditions.

C3: The number of entries that satisfy current search conditions and are mentioned after entering "Check Information" mode.

C4: The number of key attribute uttered during the dialogue.

C5: The number of key attributes that match current search conditions and are mentioned during the dialogue.

C6: Existence of key attributes mentioned during "Check Information" mode.

C7: C3/C2

C8: C5/C2

C9: Whether the content word is a key attribute and has been mentioned.

C10: Whether the content word is a key attribute and matches current search conditions.

C11: Whether the number of entries matched for current search conditions is 0 or not.

C12: Whether the number of entries matched for current search conditions is 1 or not.

C13: Depth of the current tree structure.

C14: Depth of the node overwritten by the content word. (If the content word corresponds to new search condition, (C13+1) is used.)

C15: Average depth of nodes that have same attribute as the content word. (If there are no such nodes, (C13+1) is used.)

C16: The number of children of the node overwritten by the content word.

C17: Whether the content word has already been included in the current search conditions.

C18: C14/(max_depth+1).

C19: (Current_depth+1)-C14.

C20: C15/(max_depth+1).

C21: (Tree_Depth+1)-(C15).

Figure 3.7: Features based on proposed models.

$CF(s|F_i, w_i)$ of each word w_i in the utterance, as follows.

$$S = \arg\max_s \sum_i CF(s|F_i, w_i)$$

Here, s indicates a dialogue act and F_i indicates the features of the decision tree.

2. Select content words that the system accepts. The selection procedure is conducted independently for each content word w_i by using the confidence of rejection $R_i = CF(reject|F_i, w_i)$, as follows.

   - If $CF(S|F_i, w_i) \geq R_i$, w_i is accepted.
   - Otherwise, w_i is rejected.

   Note that if the confidence of rejection of the content word is low ($R_i < \alpha$), and if at the same time $CF(S|F_i, w_i) \neq 0$, the system confirms with the user whether the system should accept that content word in order to prevent a mistaken rejection of a true content word. We set $\alpha = 0.9$ in our experiment.

3.5 Experimental Evaluation

3.5.1 Collection of Evaluation Data

We implemented our proposed dialogue system in a restaurant database search for collecting dialogues. In the restaurant database, one key attribute, "Restaurant Name", and 11 non-key attributes, such as "Food Type" or "Address", are defined[4]. The number of entries in the database is 1,217.

We used Julius [73] as a speech recognizer. A new language model was made by combining two existing language models: one trained from hand-crafted putative utterances (vocabulary size: 2,185) and one trained from a large corpus prepared for a gourmet recipe domain (vocabulary size: 19,447) [74]. The combination ratio was set to 9:1, resulting in a total vocabulary size of 21,565.

Before collecting dialogue data from actual participants, we performed a preliminary collection of dialogue data from six initial participants in our laboratory. This data yielded 748 content words that we manually labeled for training the decision tree in the proposed

[4]All attributes are listed in Table 3.1.

language understanding module. This trained decision tree was then used in the system for collecting dialogues from actual participants. We used C5.0 [72] as the decision tree.

System responses are displayed on the console and output as synthesized speech simultaneously. If more than eight entries match the current search conditions, the system outputs only the number of entries. If there are fewer than eight entries, the system outputs the number of entries and corresponding restaurant names.

We used our system to collect dialogues from 20 participants who had no experience using any dialogue system. Participants were first asked to read a brief explanation and examples of utterances that the system can understand. They were then asked to use the system for about five minutes to get accustomed to it. Next, participants were shown various dialogue scenarios such as "You want to eat Japanese food, but you don't have any cash, only a VISA card". After reading these scenarios, participants were asked to dialogue with the system until they found a good restaurant that satisfied their demands. Dialogues were collected under three scenarios that we prepared beforehand and one free scenario that each participant invented on their own.

3.5.2 Details of Evaluation Data

From the procedure described above, we collected 3,015 utterances (151 utterances per participant, 38 utterances per dialogue), which included a total of 2,803 content words. Word accuracy was 78.9%. The number of utterances and content words for each dialogue act are listed in Table 3.2. Note that eight utterances requested by users contained the dialogue acts of "Addition of Search Conditions" and "Deletion of Search Conditions" simultaneously. Such utterances are counted for each dialogue act in Table 3.2. The "Others" label in Table 3.2 indicates that either the dialogue act of the utterance was "Positive Acknowledge", "Negative Acknowledge", "Deletion of All Search Conditions", or "Undo" or that the utterance was out-of-task. The number of utterances in the former case was 342 and that in the latter was 161. From Table 3.2, we can see that "Addition of Search Conditions" and "Check Information" had almost the same number of utterances, and that they each had about 20% of speech recognition errors. This indicates the importance of rejecting speech recognition errors or resolving disambiguation for correct language understanding.

A decision tree trained using all evaluation data with all features is shown in Figure 3.8. The features based on the proposed models—for example, the "current mode

Table 3.2: Number of utterances and content words per dialogue act.

	Addition of Search Conditions	Check Information	Deletion of Search Conditions	Others	Total
Utterances	1,220	1,013	279	503	3,015
Actual content words	1,388	1,253	307	-	2,948
Correctly recognized content words	1,133	1,037	244	-	2,414
Recognized content words	1,279	1,177	287	60	2,803

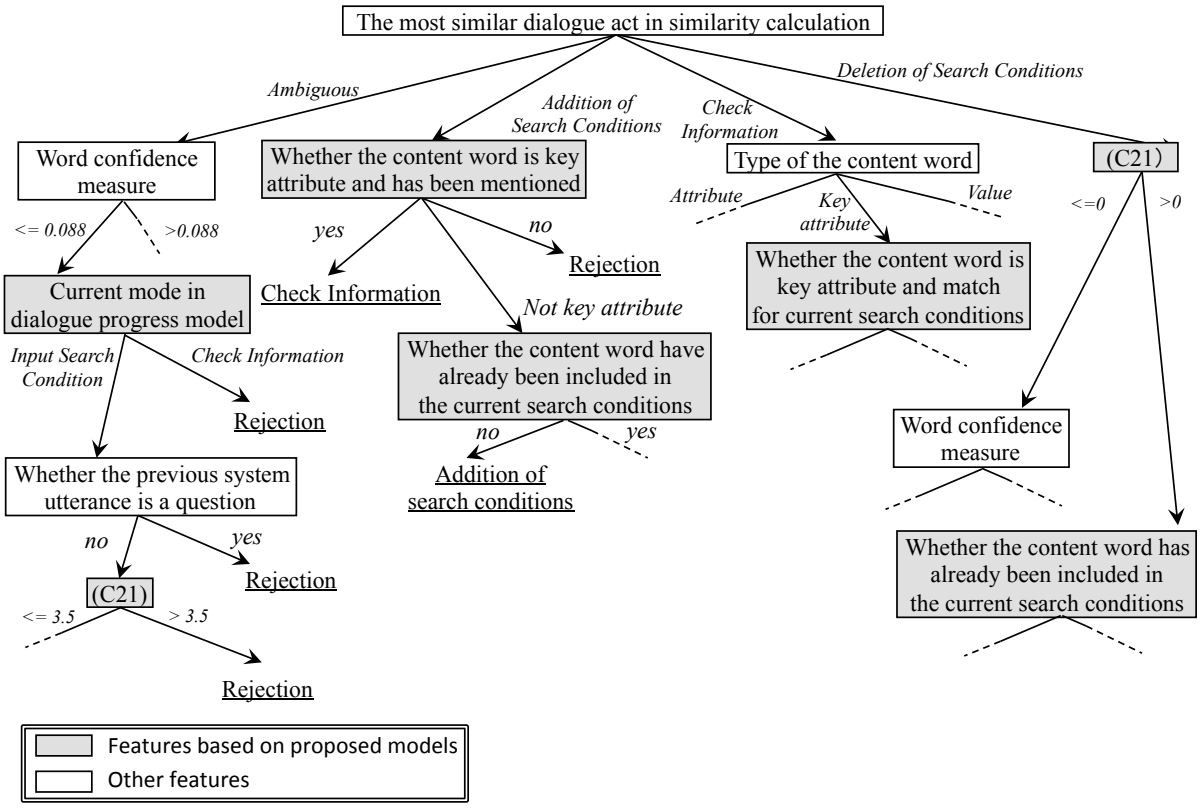


Figure 3.8: Excerpt of decision tree trained using dialogue data.

in the dialogue progress model" and the normalized tree depth in the history structure model (C21)— appeared in the upper part of the tree, indicating the importance of those features.

We present an example from the collected corpus to show how this decision tree was able to manage correct language understanding. A user uttered "Izakaya Mumon no jusho wo onegai shimasu (Tavern Mumon's address, please.)". However, this utterance was misrecognized as "Izakaya wo mon no jusho wo onegai shimasu (Tavern wo mon ' s address, please)". Note that while "Izakaya Mumon" is a restaurant name, "wo mon"

were misrecognized words and "Izakaya (Tavern)" was incorrectly understood as a " Food Type " search condition[5]. At that time, the calculated similarity of the "Addition of Search Conditions" dialogue acts were same as those of "Check Information", and so the similarity calculation module output the label "ambiguous". Two content words, "Izakaya (Tavern)" and "Jusho (Address)", were extracted from the speech recognition result[6]. In addition, the confidence measure of "Izakaya (Tavern)" was 0.65, which was relatively high despite having to prevent the word from being accepted as a search condition. The current mode was "Input Search Condition" and the previous system utterance was not a question. The depth of the search condition overwritten by the word "Izakaya (Tavern)" was relatively small, and as a result the feature (C21) was 4, which is a relatively high value. From these observations, the decision tree in Figure 3.8 decided that the word "Izakaya (Tavern)" should be rejected as a content word, which is a desirable result for the language understanding module. At the same time, the content word "Jusho (Address)" was classified as "Check Information". By integrating the outputs for "Izakaya (Tavern)" and "Jusho (Address)", our language understanding module could determine that the dialogue act of the utterance was "Check Information" with one content word "Jusho (Address)". Note that these are the best possible results extracted from the speech recognition result, which enables dialogue systems to respond to users requests such as "Restaurant name, please.".

3.5.3 Evaluation of Language Understanding Accuracy

To evaluate the proposed method, we compared the three methods below.

Method 1 (Baseline): In this method, a dialogue act is selected as that which outputs the highest similarity between a speech recognition result and putative utterances for that domain. A content word is accepted if its word confidence is higher than a certain threshold.

Method 2: This is the proposed language understanding method with a decision tree without features based on the proposed context models (Figure 3.7). Acceptance or rejection of a content word is decided according to the proposed method.

[5]In the restaurant database that we used, both categories of food and categories of restaurant are included in the "Food Type" attribute.

[6]Extraction of content words was implemented as simple string matching with attributes and values in the restaurant database.

Table 3.3: Language understanding accuracy (F-measure) for each content word in the restaurant system.

	Method 1	Method 2	Method 3
Addition of Search Conditions	0.926	0.907	0.903
Check Information	0.945	0.953	0.949
Deletion of Search Conditions	0.815	0.730	0.857
Rejection	0.100	0.368	0.550
Total	0.809	0.834	0.867

Table 3.4: The number of language understanding errors for each content word in the restaurant system.

	Method 1	Method 2	Method 3
Language Understanding Errors	536	465	373

Total content words: 2,803

Method 3: This is the proposed language understanding method with a decision tree with all features. Acceptance or rejection of a content word is decided according to the proposed method.

The language understanding accuracy for each content word is listed in Table 3.3 and the number of errors for each content word is listed in Table 3.4. We used F-measure[7] to evaluate the language understanding accuracy. For method 1, we evaluated 20 different thresholds and then chose the best one, which was 0.05. If multiple dialogue acts output the same similarity in the similarity calculation, we heuristically selected "Addition of Search Conditions", resulting in a dialogue act accuracy of 78.2% in such ambiguous cases. For methods 2 and 3, we conducted 10-fold cross validation for an open evaluation of the decision tree. We split the dialogue data into 10 folds according to participants, i.e., the dialogue data of 18 participants were selected for training the decision tree and that of 2 participants were used for evaluating the proposed method. We iterated this procedure 10 times while changing the evaluation data. Twenty different pruning parameters of the decision tree were tested and we selected the one that best minimized the language understanding errors.

In the upper part of the decision tree that was trained in method 2, we found features such as "the most similar dialogue act in the similarity calculation" or "word confidence measure", which indicates that method 2 utilized similar information as method

[7] F-measure=2·Recall·Precision/(Recall+Precision).

1. Method 2 also utilized other utterance-based features such as the second-most similar dialogue acts (S4) and general context features such as "whether the content word had been denied". By incorporating these features, the rejection accuracy of method 2 was greatly improved. Although the accuracy of "Addition of Search Conditions" and "Deletion of Search Conditions" was slightly degraded due to rejection misrecognitions, the overall accuracy was improved by 2.5 points.

Compared to method 2, method 3 could further reduce 92 language understanding errors, which corresponds to a 19.8% (=92/465) relative error reduction. Total F-measure was improved 3.3 points. The difference between methods 2 and 3 was in the utilization of features based on the proposed context models, and so these results highlight the effectiveness of our models. Rejection accuracy was especially improved, showing 18.2 points of improvement in the F-measure. In the decision tree of method 3, the top feature was "the most similar dialogue act in the similarity calculation", the same as method 2. However, in this case many features based on the proposed context models appeared in the upper part of the decision tree. For content words corresponding to search conditions, "current mode in dialogue progress model" and features based on the history structure model (e.g., C14) were frequently found. For attributes (including key attributes), "The number of entries that satisfy current search conditions", "Whether the content word has already been included in the current search conditions", and "Whether the content word is a key attribute and has been mentioned" appeared in the upper part of the tree.

While method 3 had 163 fewer errors than the baseline method 1, there were still 373 words that caused language understanding errors. Errors that occurred most frequently in the evaluation happened when the language understanding module accepted a speech recognition as "Addition of Search Conditions"; there were 133 such cases. Among these, 31 speech recognition errors were the substitution of values related to the same attribute (e.g., "Shi-jo" and "Shichi-jo"). The rejection of such errors was difficult even for the proposed method because the proposed features cannot capture context among the same attribute. In addition, it was difficult for the proposed method to utilize context information at the beginning of a dialogue. We found 18 errors at the beginning of a dialogue.

For both methods 2 and 3, the top feature on the decision tree was "the most similar dialogue act in the similarity calculation". We therefore evaluated the relation between the language understanding errors and the number of putative utterances, by which the accuracy of the similarity calculation is mainly affected. The results are shown in Figure 3.9.

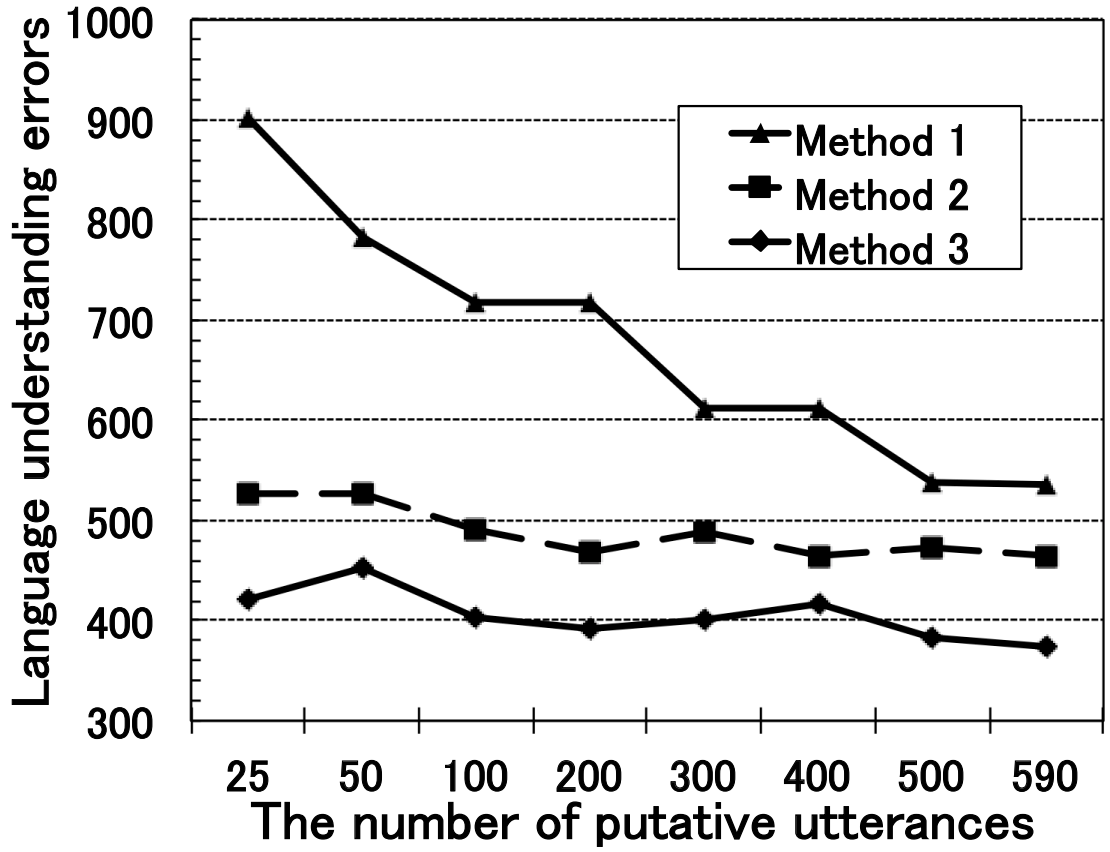


Figure 3.9: The number of putative utterances and language understanding errors.

First, we found that the language understanding errors of method 1 were dramatically increased when reducing the number of putative utterances. In contrast, methods 2 and 3 were not so sensitive to the number of putative utterances, and only a slight increase of language understanding errors was observed when the number of putative utterances was reduced. This indicates that the features introduced in Section 3.4.2 could robustly complement the information in the language understanding procedure, even when the similarity calculation was not perfectly accurate.

3.5.4 Evaluation of Domain Exchangeability of the Proposed Method

The proposed method was designed to function independently of database domain. To evaluate the domain exchangeability of our method, we constructed a hotel database

Table 3.5: Language understanding accuracy (F-measure) for each content word in the hotel system.

	Method 1	Method 2	Method 3	Method 4
Addition of Search Conditions	0.917	0.940	0.964	0.930
Check Information	0.978	0.967	0.983	0.990
Deletion of Search Conditions	0.822	0.711	0.756	0.933
Rejection	0.287	0.318	0.504	0.527
Total	0.888	0.890	0.926	0.924

search system and used it to collect dialogues. The total number of entries in the hotel database was 2,004. The database consisted of the key attribute "Hotel Name" and seven other secondary attributes: "Hotel Type", "Address", "Budget Max", "Budget Min", "The Number of Rooms", and "Other Facilities". Among the seven attributes, the first four are similar to those of the restaurant database but the latter three are not. We used an n-gram language model with 6,953 vocabulary items for the speech recognition module. Note that we used the proposed language understanding module with a decision tree trained using the restaurant database we described earlier.

We used this system to collect dialogue data from 10 participants. The collection procedure was same as described in Section 3.5.1. Overall, we collected 1,271 utterances containing 1,426 content words. Word accuracy in this experiment was 83.2%.

We then used this data to evaluate the language understanding accuracy. The same three methods as in Section 3.5.3 were evaluated, along with an additional " method 4".

- **Method 4:** This is the proposed method with a decision tree trained using dialogue data collected in Section 3.5.1, which contained 2,803 content words. All proposed features are used in the decision tree.

As in Section 3.5.3, methods 2 and 3 were evaluated by 10-fold cross validation, with each fold consisting of the dialogue data of one participant. While we tested 20 different pruning parameters for the decision tree and selected the one that best minimized the language understanding errors for method 2 and 3, for evaluating method 4, we used the default value of C5.0.

The results are listed in Table 3.5. Compared to method 2, method 3 consistently improved the language understanding accuracy even in the hotel domain, which indicates that the proposed method can function well independently of the type of database. More-

over, the language understanding accuracy of method 4 was almost the same as that of method 3, which indicates that after a decision tree has been trained on one corpus, it can be applied for other systems on other domains. Normally, collecting sufficient dialogue data with accurate reference labels requires a lot of time and money, so this domain exchangeability feature is quite valuable.

3.6 Summary

We summarize this chapter as follows.

- We proposed a new language understanding method that incorporates the dialogue contexts of dialogues in a database search task. The proposed method was able to reject speech recognition errors and resolve ambiguity by using context information.
- To model the contexts of dialogues in the database search task, we proposed two new context models: a dialogue progress model and a history structure model. The dialogue progress model is a context model that represents the progress of dialogue in a database search task with two general modes. The history structure model is a context model that manages search conditions by using a tree structure.
- Experimental evaluation indicated that the proposed method could increase language understanding accuracy in both a restaurant search system and a hotel search system. More importantly, the proposed method trained using the dialogue data collected in the restaurant search system could be used for the hotel search system without any configuration. This means it can be applied for other systems on other domains.

Chapter 4

Domain-Extensible Spoken Dialogue System with Robust Domain Selection Method

The previous chapter described our general-domain spoken language understanding method for spoken dialogue systems in a database search task. The proposed method can be used regardless of the background database, and therefore it is useful for developing a broad range of spoken language understanding modules. In this chapter, we focus on integrating these spoken dialogue understanding methods to develop a system that can respond to any type of utterance.

4.1 System Overview

As described in Section 1.3, there are centralized and distributed architectures. In this study, we used the distributed architecture and developed it for robot interactions [13]. An overview of the system, which consists of domain experts and a central manager, is shown in Figure 4.1. Domain experts are responsible for handling each dialogue in the associated domain. The central manager accepts user utterances, selects which expert should respond to these utterances, and replies to the user using responses produced by the selected domain expert. In our system, some information slots can be shared among domain experts through the domain manager according to specific protocols. Every communication between the domain experts and the central manager is done according to these protocols, so system developers can update or create new domain experts without bothering with outside domain knowledge.

In the distributed architecture, how to reply to the user utterances is totally the re-

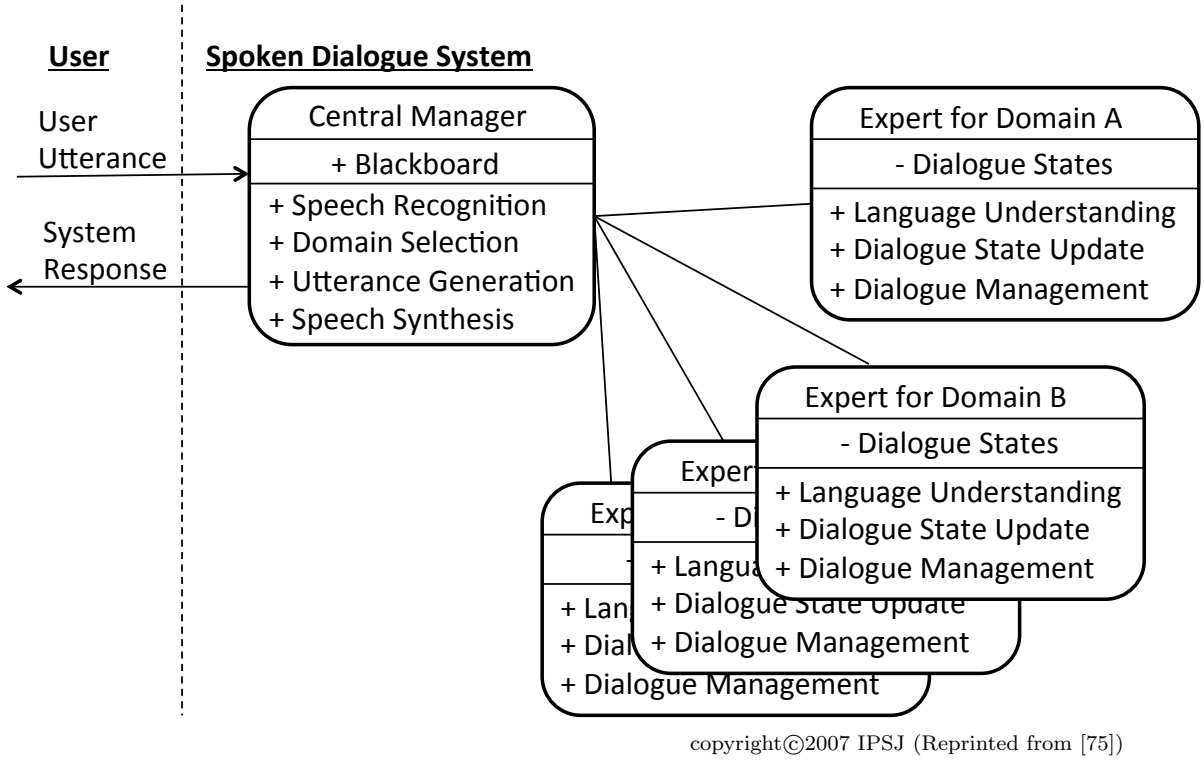


Figure 4.1: Overview of distributed spoken dialogue systems.

sponsibility of the domain experts, so the domain manager's selection of which experts to use for which questions has the utmost importance. We call this procedure **domain selection**. Note that to make a distributed architecture that has both domain extensibility and domain maintainability, the domain selection procedure must also have such properties.

There are two requirements for the domain selection procedure to have domain extensibility and domain maintainability. First, the domain selection procedure should use only domain-independent information. If domain-dependent knowledge is used to select domain experts, it becomes difficult to update domain knowledge because every time domain knowledge is updated, the domain selection procedure has to be modified to incorporate the information. Second, the domain selection procedure shold not be the one that can be applicable only for known domains. If it can handle only known domains, it becomes difficult to add new domain experts because every time a new domain is added the domain selection has to be modified to accept the new expert. We propose a new domain selection procedure that satisfies the above two requirements. Before describing our method, we provide a brief survey of related studies in the next section.

4.2 Related Studies on Domain Selection

4.2.1 Domain Selection Based on User Utterances

The simplest method of domain selection is having users explicitly utter the desired domain name, such as "bus" or "restaurant". This method can correctly detect the domain a user wants if the number of domains is small and the user perfectly understands the domain names that the system can understand. However, this method makes dialogue unnatural and verbose, and it also imposes a heavy burden on users in that they have to learn the boundary of each domain knowledge item that the system developer arbitrarily defined. For example, if a user wants to find out how to get to Kinkaku-ji Temple by bus, it is difficult to know if it is correct to say "temple" or "bus". This problem stems from the knowledge gap between users and system developers. It frequently occurs when the user requirement is related to multiple domains, and the problem becomes worse when the dialogue system handles a larger number of domains.

To reduce the user burden of explicitly selecting domains, domain estimation techniques from natural user utterances have been studied. In [24], speech recognizers are prepared for each domain, and the domain that outputs the highest likelihood when recognizing the user utterance is selected as the domain of that utterance. [7] proposed using a vector-space model that is frequently used for document retrieval. In this model, many documents (or utterances) are prepared for each domain, and the domain that has the highest number of documents similar to the user utterance is selected. [76] proposed using a support vector machine instead of similarity on a vector space model. The problem with these techniques is that they are all based on scores calculated using the speech recognition results of user utterances and do not utilize dialogue contexts. This means they easily select incorrect domains with trivial speech recognition errors. For example, with U3 in Figure 4.2, the user uttered "3000 yen (san zen yen)" for the request in the restaurant domain. However, the speech recognizer output "sanzen temple (san zen in)". In such a situation, the utterance-based domain selection methods will incorrectly select a sight-seeing domain and therefore provide the user with an incorrect response (S3 (NG) in Figure 4.2). The user is confused by such a response because the dialogue contexts of the restaurant domain before that utterance have been totally ignored.

U1: **Please tell me about Ginkaku-ji Temple.** (Sightseeing)

S1: Ginkaku-ji Temple is ...

U2: **Is there a Japanese restaurant nearby?** (Restaurant)

S2: There are 5 Japanese restaurants in the Kita-Shirakawa area. Do you have any demands?

U3: **3,000 Yen** (Restaurant)

⟨⟨*Speech Recognition Results*⟩⟩

1best: Sanzen-In Temple.

2best: 3, 000 Yen. (san zen yen)

3best: Ranzen.

↓

S3 (NG): Sanzen-In Temple is ... (Sightseeing)

S3 (OK): Budget 3,000 Yen is added to the search conditions.
I found two restaurants. (Restaurant)

Figure 4.2: Example dialogue in a multi-domain spoken dialogue system.

4.2.2 Domain Selection based on the Previous Domain

It is important to utilize information from the previous domain transition in order to consistently select domains, especially if the speech recognizer is not reliable. Various studies have utilized the information of a previously estimated domain for the domain selection procedure. In [6, 8], domains are selected based on the likelihoods of speech recognition results and a predefined positive value is added to the likelihood of the previously estimated domain to make a bias that keeps that domain. In [10], a domain is selected based on the first utterance and then kept until the user finishes the task in that domain. Assuming that the previously estimated domain is correct, these methods can reduce incorrect domain transitions caused by speech recognition errors or language understanding errors. For example, in Figure 4.2, the utterance U2 was correctly estimated as a restaurant domain, and so the system would be able to produce a correct response such as S3 (OK) if it has the ability to keep previous domains.

However, these methods are problematic in that they repeatedly select an incorrect domain if the previously estimated domain is incorrect. Such an example is shown in Figure 4.3. In this example, the user uttered an utterance U1, which should have been responded to by the sightseeing expert but was actually responded to by the bus expert because of a speech recognition error. The user then tried to correct the domain from bus to sightseeing, but the incorrect bus domain was again selected (S2 (NG) in the figure) by the bias to keep the previous domain. This problem is caused by trusting in the previous domain without checking its reliability. In our work, we take into account the histories of the previous domain in order to check the reliability of the information and to avoid selecting incorrect domains repeatedly.

4.2.3 Detection of Domain Selection Errors

The studies described above are based on the speech recognition results of a user utterance or the information of a previously selected domain. Here, let us describe another situation in which such information is not sufficient. An example is shown in Figure 4.4. In this dialogue, the user said in U1 an utterance about the hotel domain, but the restaurant expert responded because of the speech recognition error. Then, in U2, the user tried to correct the domain from restaurant to hotel, but the speech recognizer again produced errors, thinking the utterance was related to the bus domain. In this situation, the

U1: **Address of Ginkaku-ji Temple.** (Sightseeing)

(Mis-recognized as "Bus from Ginkaku-ji Temple.")

S1: Where do you want to go from Ginkaku-ji Temple?

U2: **Place.** (Sightseeing)

⟨⟨*Speech Recognition Results*⟩⟩

1best: Address

2best: Buses

↓

S2 (NG): Where do you want to go from Ginkaku-ji Temple?

S2 (OK): Address of Ginkaku-ji Temple is ... (Sightseeing)

Figure 4.3: Example dialogue in which the constraint to keep previously selected domain does not work.

correct domain to be selected is neither the previously selected domain (restaurant) nor the domain estimated from the speech recognition result (bus). If the system can identify such a confusing situation, it can avoid continuing incorrect dialogue such as utterance S2 (NG) and respond with a correct utterance such as utterance S2 (OK).

In the situation above, both the previously estimated domain and the speech recognition result have been incorrectly recognized. To identify such a situation, the system needs to detect not only speech recognition errors but also domain selection errors. While many studies have focused on the detection of speech recognition errors (such as [58–60]) or language understanding errors (such as [66, 77]), so far as we know there has been no research on detecting domain selection errors. In our study, we estimated the reliability of selected domains by incorporating the information of dialogue history to detect domain selection errors.

4.2.4 Domain Selection using Dialogue History

As stated above, it is necessary to know the reliability of information (i.e., if there are speech recognition errors and domain selection errors) in order to accurately estimate domains. If the previous domain has been estimated reliably, it would be logical to keep

U1: **Kyoto...** (Hotel)

(Misrecognized as "Kyoto cuisine".)

S1: Searching for Kyoto cuisine restaurants. 50 restaurants are found.

U2: **No, I want the address of Kyoto Grand Hotel.** (Hotel)

(Misrecognized as "Bus from Kyoto Grand Hotel.")

↓

S2 (NG): Where do you want to go from Kyoto Grand Hotel by bus?

S2 (OK): Would you like to talk about restaurants or bus information?

Figure 4.4: Example dialogue in which both previously selected domain and most probable domain based on speech recognition results are incorrect.

this domain. Similarly, if the speech recognition result is reliable, the domain estimated from the speech recognition result should be used. If neither the previously estimated domain nor the speech recognition result are reliable, the system should take some action to recover, as with S2 (OK) in Figure 4.4. As we already have some information about the reliability of speech recognition results, what we especially need is a way of determining the reliability of previously estimated domains.

In our study, we propose utilizing dialogue history to estimate the reliability of selected domains. For example, if a user makes a positive acknowledge (such as "yes" or "correct") in the dialogue, the selected domain is considered reliable. As another example, if the dialogue states are not changed over several turns, the selected domain might be incorrect. We use dialogue history in this way to estimate the reliability of selected domains. Incorporating the information enables us to robustly select domains even when there were speech recognition errors or previous domain selection errors.

We also utilize the dialogue states in the selected domain. For example, if the user has just accomplished a task in the selected domain, he or she is expected to switch from the current domain to another domain according to the change of topic, and it would be rare to switch domains before accomplishing any task. Such information could be used for domain selection if the status of the task is defined in a uniform way. In [6], the task status is defined as one of three states—"waiting (beginning)", "middle", and

"accomplished"—and the system uses this information to manage the domain switching. In [10], the authors define the task as "middle" or "accomplished" and do not permit changing the domains when the task status is "middle". There is, however, another type of dialogue that cannot be defined as such a type of task status.

In [67], the authors proposed classifying the tasks of spoken dialogue systems into three types: the slot filling task, the database search task, and the explanation task. The slot filling task can be effectively managed by using system-initiative dialogue because the required slots to accomplish the task are all defined before the dialogue begins. Therefore, in the slot filling task, the accomplishment of the task can be defined explicitly. In contrast, in the database search task, the accomplishment of the task cannot be defined explicitly because users normally have different needs when they access a database. For example, in the restaurant database search task, one user might want to search for an inexpensive restaurant and another might want a Japanese restaurant. The system cannot know the required search criteria before the dialogue and therefore cannot normally determine whether the dialogue objective has been accomplished or not. Instead of using the "accomplished" status, we use two task statuses for the database search task, "input search condition" and "check information", described in Chapter 3. We used these two task statuses for the experts related to the database search task and did not consider the explanation task.

4.3 Robust Domain Selection Method with Domain Extensibility

4.3.1 Definition of Domain Selection

In this section, we explain our definition of domain selection, which is the key means by which our domain selection method ensures domain extensibility. We define the domain selection as a task to select domains from the three options listed below.

- **Option (I)** - The previously selected domain.
- **Option (II)** - The most probable domain estimated from the speech recognition result.
- **Option (III)** - Another domain.

In this definition, option (I) corresponds to domain selection based on the previous domain and option (II) corresponds to domain selection based on user utterance. As such, these definitions include the conventional domain selection methods described in Section 4.2. We came up with option (III) for situations in which both the previously selected domain and the most probable domain are not reliable.

In our method, a classifier that determines which of the above three options to use is trained by using many dialogue features[1]. Utilizing both information about speech recognition errors and domain selection history gives us the ability to consistently keep the previous domain even if speech recognition errors occur. For example, in Figure 4.2, a user secondly spoke an utterance related to the restaurant domain in U2 and then spoke a third utterance about restaurants. However, a speech recognition error occurred because of a similar pronunciation word. At that time, if the domain selector can select the previously selected domain (option (I)), our system can correctly respond to the user as with S3 (OK) in Figure 4.2.

Option (II) indicates that a user has changed the domain. Here, two cases are possible: one, the user has actually changed the topic, and two, the previously selected domain is incorrect and the user has requested the correct domain. In both cases, it is important to estimate the reliability of the speech recognition results in order to correctly select option (II).

Option (III) of our definition gives us the ability to detect a situation in which both the previously selected domain and the most probable domain estimated from the user utterance are incorrect, as in the example in Figure 4.4. Identifying such a situation is more difficult than identifying speech recognition errors because the domain selection method needs to consider the possibility of both speech recognition errors and domain selection errors. For taking into account the domain selection errors, we used many features related to dialogue histories, some of which are discussed in Section 4.2.4.

Note that our definition of domain selection is not designed to explicitly select a domain name, as is done in domain selection methods based on user utterances (Section 4.2.1). If the domain selection method is designed to select a domain name, the domain selector no longer has domain extensibility because every time a new domain expert is added to the system, the domain selector needs to be modified to accept it.

[1]The decision tree classifier used in our evaluation is described in Section 4.4.

In contrast, our definition of domain selection does not rely on domain names but rather on the relative relationship among domains in a time series, which means it can be used without any modification, even if a new domain expert is added to the system. This property makes our domain selection method both domain extensible and domain portable, as has been experimentally verified in a work [78].

4.3.2 Feature Set for Domain Selection

This section describes the features we used to perform the domain selection defined in the previous section. We utilized not only information extracted from a speech recognizer but also that obtained from dialogue history for the robust estimation of dialogues. The feature set consists of two groups: **features representing the reliability of option (I)** and **features representing the reliability of option (II)**. We felt that because option (III) is a complementary class of options (I) and (II), it would be sufficiently represented by the above features. We prepared 32 features [79] and selected the most valuable features from among them. The feature selection procedure is described in Section 4.3.3 in detail. Here, the 23 selected features are listed in Figure 4.5 and Figure 4.6.

In our domain selection method, option (I) is selected if the previously selected domain is reliable and the domain should be continued. The 14 features to estimate such conditions are listed in Figure 4.5. We used the number of positive acknowledges (such as "yes" or "correct") as a positive clue that the selected domain is reliable (I_1). We also used the number or the ratio of slots that are changed during the dialogue as a clue of the reliability of that domain (I_5,I_7). For example, if no slots are changed during the domain, this could indicate an incorrect domain that the user did not actually require. We used the task status to represent the continuity of the dialogue. For example, if a task is accomplished in a particular domain, it would be natural to change the domain, no matter how reliable the previous domain is. As described in Section 4.2.4, the representation of task status is different depending on the task type, i.e., slot filling task or database search task. For a slot filling task domain such as a weather checking domain, we used the status "middle", which represents a situation in which required slots are not fulfilled, and the status "accomplished", which represents a situation in which all required slots have been fulfilled. For a database search domain such as a restaurant search domain, we used the statuses "input search condition" and "check information", both of which we described in Chapter 3. We also used the dialogue state assuming the selection of option (I). For

I_1: The number of positive acknowledges during the domain related to option (I)

I_2: The number of negative acknowledges during the domain related to option (I)

I_3: Whether or not some of the tasks in the domain of option (I) have already been accomplished (for database search tasks, whether or not some information in the domain of option (I) has already been offered)

I_4: Whether the domain had been selected before the domain of option (I) was selected

I_5: The number of slots changed during the domain of option (I)

I_6: The number of turns during the domain of option (I)

I_7: The ratio of slot changes ($= I_5/I_6$)

I_8: The ratio of negative acknowledges among all acknowledges ($= I_2/(I_1 + I_2)$)

I_9: The ratio of negative acknowledges among all user requests ($= I_2/I_6$)

I_{10}: Task status of the domain of option (I)

I_{11}: Task status after selection of option (I)

I_{12}: Whether or not the user utterance is understood as a negative acknowledge by the domain expert of option (I)

I_{13}: The number of slots changed when accepting the user utterance by the domain expert of option (I)

I_{14}: A posteriori probability of the speech recognition results accepted by the domain expert of option (I)

Figure 4.5: Features representing the reliability of option (I).

example, we used the number of slots that have changed if option (I) is selected (I_{13}). If I_{13} is small, it might be a mistake to select option (I) and continue the dialogue in that domain, because a small I_{13} indicates that the domain experts cannot understand the user utterance. We also used a posteriori probability of the speech recognition results related to the target domain (I_{14}) as well as the traditional domain selection methods discussed in Section 4.2.1.

The features prepared to represent the reliability of option (II) are described in Figure 4.6. We again used features obtained from the dialogue history. Similar to the features for option (II), the task status or the number of slots changed after selecting option (II) is utilized (II_1, II_3). We also incorporated the intuition that if the selection of option (II) invokes a large change of information states, the probability of domain selection error is

II_1: The task status after selecting option (Ⅱ)

II_2: Whether or not the user utterance is understood as a negative acknowledge by the domain expert of option (Ⅱ)

II_3: The number of slots changed after selecting option (Ⅱ)

II_4: The number of shared slots after selecting option (Ⅱ)

II_5: Whether the domain had been selected before the domain of option (Ⅱ) was selected

II_6: A posteriori probability of speech recognition result that the domain expert of (Ⅱ) accepts

II_7: Mean of confidences of words in the speech recognition result that the domain expert of (Ⅱ) accepts

II_8: Difference between acoustic scores of speech recognition results that the domain experts of options I and (Ⅱ) accept

II_9: Ratio between word confidence means of speech recognition results that the domain experts of options I and (Ⅱ) accept

Figure 4.6: Features representing the reliability of option (Ⅱ).

high. To represent this intuition, we used the number of shared slots changed after the domain expert of option (Ⅱ) accepts the user utterance (II_4). The large II_4 indicates that, assuming option (Ⅱ) is correct, the user changes not only the domain but also shared slots. Such radical change of information states would not frequently happen, which suggests that the selection of option (Ⅱ) is incorrect. We used the a posteriori probability of speech recognition results as well as conventional studies. In addition, we used the ratio of speech recognition results accepted by the domains of options (I) and (Ⅱ) to represent the confidence of the speech recognition results.

4.3.3 Feature Selection for Domain Selection

In this study, we constructed a classifier that determines which of the three options defined in Section 4.3.1 to use by utilizing the features described in Section 4.3.2. Too many features typically results in overfitting of the classifier, so we conducted a feature selection procedure, as described below. Note that the explanation below assumes there is enough training data to construct a classifier and enough evaluation data to measure accuracy. In this study, we used cross validation because there was not so much dialogue

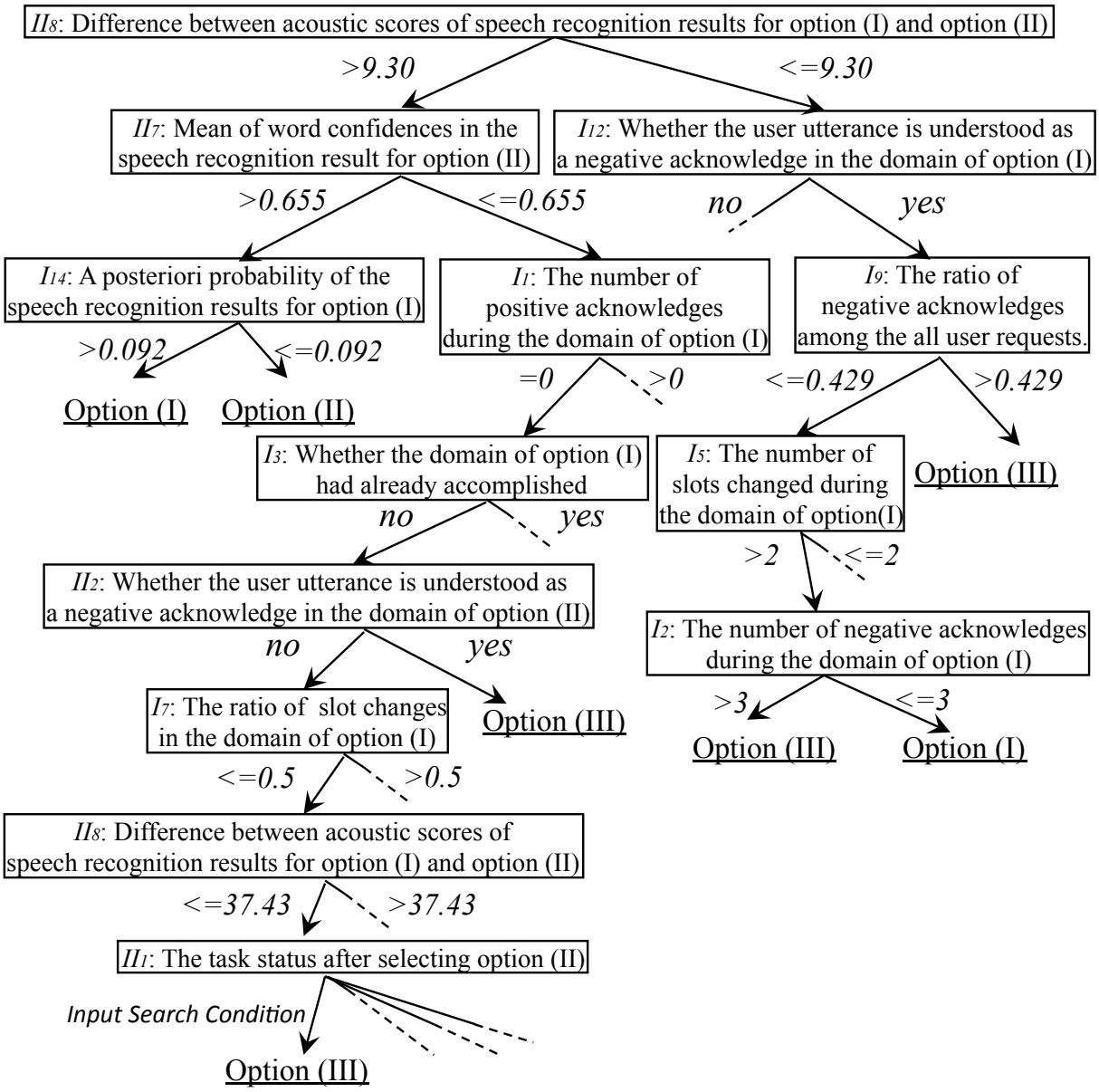


Figure 4.7: Decision tree constructed from dialogue data.

data.

1. Features described in [79] are set as $\boldsymbol{F}$.

2. Execute following (a) and (b) for each feature $a \in \boldsymbol{F}$.

   (a) Train a classifier $M_{F\backslash a}$ by using features $\boldsymbol{F}\backslash a$, which are $\boldsymbol{F}$ except a.

   (b) Evaluate the accuracy $V_{F\backslash a}$ of the classifier $M_{F\backslash a}$ by using evaluation data.

Table 4.1: Specifications of each domain.

Domain	Task Type	Vocabulary Size of Speech Recognizer	Number of Slots
Restaurant	DB	1,562	10
Hotel	DB	741	9
Sightseeing	DB	1,573	4
Weather	SF	87	3
Bus	SF	1,621	3
Total	-	7,373	-

DB: Database Search Task, SF: Slot Filling Task

3. Calculate $\tilde{a} = \text{argmax}_{a \in F} V_{F \backslash a}$.

4. If accuracy of the classifier improves or does not change by removing some of the features, then $\boldsymbol{F} \leftarrow \boldsymbol{F} \backslash \tilde{a}$, and return (2).

5. Stop the iteration.

4.4 Experimental Evaluation

4.4.1 Construction of Evaluation Data

We developed a multi domain spoken dialogue system to collect evaluation data for various domain selection methods. We constructed five experts for five different domains: restaurant search, hotel search, sightseeing search, weather information, and bus information. The details of each domain are shown in Table 4.1. While the base systems are implemented in Java, experts can be written in any programming language as long as they follow the specified protocol. In fact, while the restaurant, hotel, sightseeing, and weather experts are written in Java, the bus expert is written in Perl. As mentioned in Section 4.1, our system has a special protocol to share slots among experts. By using this protocol, some topics can be taken over when changing the domains. In this study, we used this protocol to share the slots related to place information. The granularity of place slots is differently designed for each domain expert, so we shared information about latitude and longitude converted from each domain slot.

In this study, we used Julian [73], a grammar-based speech recognizer, to recognize user utterances. Grammars for Julian are automatically generated from grammars designed for the language understanding module of each expert. A PTM triphone acoustic model with 3,000 states, which is included in the Julius Dictation Kit [73], is used for Julian.

Each expert is designed to reply to a user utterance with implicit acknowledgement, as exemplified by S1 in Figure 4.3. We not only output a synthesized speech of the system utterance but also displayed it for users so that they did not miss what the system said.

We collected dialogue data from ten participants using this system. Each participant had an initial dialogue with the system for about 10 minutes with an easy scenario in order to get accustomed to using it. They were then asked to dialogue with the system according to a scenario that normally requires three or four changes of domain to reach its conclusion. Each participant had three dialogues with different scenarios.

When collecting the data, the system selects the domain according to the speech recognition results. The ten best speech recognition results are processed by language understanding modules by all domain experts one by one, and the domain is selected according to the highest probability of speech recognition result that the domain expert can accept. Note that when selecting the domain expert, a bias score was added to the previously selected domain to keep that domain. This bias score was set to 40 on the basis of preliminary experiments.

Through the dialogues, 2,205 utterances (221 utterances per participant, 74 utterances per dialogue) were collected. Word accuracy was 63.3%, which is relatively low because users tended to repeatedly produce the same speech recognition errors caused by out-of-grammar or out-of-vocabulary utterances. This created a kind of snowball effect in that if the word accuracy of a user was low, he or she required more turns to complete the dialogues. Of course, this severely affected the word error rate. In the collected utterances, there were 274 utterances in which the speech recognition results were a typical positive acknowledgement (e.g., "yes"). Such utterances made it very easy to select domains—just select option (I) (previously selected domain)[2]. Therefore, we tried to evaluate the accuracy of the domain selection methods without these utterances. We show the results both with and without positive acknowledgement.

All utterances are manually transcribed and labeled with one of the three options of domain selection. The labeling procedure was executed according to the protocol below.

- If a manually selected domain is the same as the previously responded domain, option (I) is labeled.

- If a domain is the same as the domain that the domain expert can accept as the

[2]In contrast, negative acknowledgements (e.g., "no") had the possibility of option (III).

most probable speech recognition result compared to other experts, option (II) is labeled. Note that if both conditions for option (I) and option (II) are satisfied, option (I) is labeled.

- Otherwise, option (III) is labeled.

As a classifier for our domain selection method, we used the decision tree classifier C5.0 [72]. An example decision tree created by the classifier is shown in Figure 4.7. We found that the top feature was the difference between the scores of speech recognition results (II_8). We also found that features relating to dialogue histories appeared higher up in the decision trees, indicating that these features are valuable. For example, we found the number of positive acknowledges during the domain related to option (I) (I_1), the ratio of negative acknowledges among all user requests (I_9), and the number of slots that changed during the domain of option (I) (I_5) in the upper part of trees.

An excerpt of dialogues that the decision tree correctly performed is shown in Figure 4.8. In this dialogue, the user first said "The temperature of tomorrow please", but this was misrecognized by the speech recognizer and the restaurant expert replied. The user then said "Weather please", but this was again misrecognized as an utterance related to sightseeing (U2). At this time, the previously selected domain was the restaurant domain (option (I)) and the best possible domain from the speech recognizer was the sightseeing domain (option (II)). In addition, the following conditions were observed:

- The difference between the scores of speech recognition results that the restaurant expert (related to option (I)) and the sightseeing expert (related to option (I)) could accept was 26.2.

- The arithmetic mean of word confidences in the speech recognition results of the sightseeing expert (related to option (II)) was relatively small: 0.64.

- During the dialogue in the restaurant domain (related to option (I)), no positive acknowledge was observed, no task was completed, and no slot was changed[3].

- If accepting option (II), the task status in the restaurant domain was changed to "input search condition".

[3]Slots under confirmation were treated as not being changed.

U1: **Ashita no kion wo onegai shimasu.**
(The temperature of tomorrow please.)

—Misrecognized as "Pasta wo onegai shimasu(Pasta please)"—

S1: Food type ga pasta no restaurant wo kensaku shitemo yoroshii desuka?
(Would you like to search for restaurants that serve pasta?)

U2: **Tenki wo onegai shimasu.**
(Weather please.)

—Misrecognized as "Enfukuji wo onegai shimasu(Enfukuji-temple please."—
—Then, our domain selection method selected option (Ⅲ).—

S2: Restaurant matawa jisha ni tsuite otazune desuka?
(Are you asking for restaurants or temples?)

Figure 4.8: Example in which option (Ⅲ) was selected.

On the basis of the conditions above, the decision tree in Figure 4.7 can correctly select option (Ⅲ). This enables the system to respond to the user with "Are you asking for restaurants or temples?", which prevents the system from continuing dialogues in incorrect domains (S2).

4.4.2 Evaluation of Domain Selection Accuracy

We evaluated the accuracy of the two domain selection methods below.

- **Baseline Method**, which selects the domain by comparing the highest probability of speech recognition result that the domain expert can accept. Note that a bias score α was added to the previously selected domain.

- **Proposed Method**, which is described above. For evaluating this method, we used 10-fold cross validation, in which each fold corresponds to one participant. A cut-off parameter of the decision tree classifier was set so as to maximize the accuracy.

For calculating the accuracy, a domain selection procedure was applied utterance by utterance and the selected result (from option (I) to option (Ⅲ)) was checked to determine if it was correct or not. If the domain selection method produced the same score for multiple options, an option was randomly selected and evaluated.

We first evaluated the accuracy of the baseline method for utterances excepting positive acknowledgements, changing the bias parameter α from 0 to 100. Note that a larger

Table 4.2: Confusion matrix of domain selection results for all utterances (baseline / our method).

Reference \ Hypothesis	(I)	(II)	(III)	Total (Recall)
(I)Previously selected domain	1,289/ 1,291	162/ 85	0/ 75	1,451 (0.89/ 0.89)
(II)The most probable domain for speech recognition results	84/99	299†/ 256†	0/ 28	383 (0.74/ 0.62)
(III)Another domain	293/ 172	78/ 42	0/ 157	371 (0/ 0.42)
Total (Precision)	1,666/ 1,562 (0.77)/ (0.83)	539/ 383 (0.52)/ (0.62)	0/260 (-)/ (0.60)	2,205 (0.712/ 0.765)

†The number of utterances that the system cannot resolve the ambiguity of the domain because of Multiple domains that have the same selection score [(17 for both the baseline method and proposed) (OK?)] method, respectively.

α leads to a stronger tendency to keep the current domain. In contrast, if α is set to 0, the baseline method is equal to the method that simply selects the domain that can accept the most probable speech recognition results. In our investigation, the number of domain selection errors was smallest when $\alpha = 35$. At that time, the number of errors was 618 and the error rate of domain selection was 32.0% (=618/1,931). Note that there were 361 errors in which the correct label was option (III), which is not considered and therefore cannot be selected by the previously proposed methods, including the baseline method.

We next evaluated the accuracy of the proposed method. Table 4.2 and Table 4.3 show the confusion matrix of domain selection results for all utterances and for all utterances excepting positive acknowledgements, respectively. In both tables, the left-most line indicates the reference labels and the top-most line indicates the selected domains. Cells on the diagonal line indicate the number of correctly selected domains and other cells indicate domain selection errors. The total accuracy in Table 4.2 is about 3% better than that in Table 4.3 because the former results include positive acknowledgements, which are easy to find in the correct domain.

Looking at Table 4.3 in detail, we first found that the total number of domain selection errors was reduced from 618 with the baseline method to 518 with the proposed method. Total accuracy increased from 68.0% to 73.2%, which corresponds to a 16.2% (=100/618) relative error reduction. We also found that our method was able to detect 158 utterances of option (III), which the baseline method could not. These results indicate that the proposed method can identify about half of the situations in which both the previously

Table 4.3: Confusion matrix of domain selection results for utterances excepting positive acknowledgements (baseline / our method).

Reference \ Hypothesis	(I)	(II)	(III)	Total (Recall)
(I)Previously selected domain	1,031/ 1,023	162/ 87	0/ 83	1,193 (0.86/ 0.86)
(II)The most probable domain for speech recognition results	78/95	299†/ 247†	0/ 35	377 (0.75/ 0.62)
(III)Another domain	283/ 161	78/ 42	0/ 158	361 (0/ 0.44)
Total (Precision)	1,392/ 1,279 (0.74)/ (0.80)	539/ 376 (0.52)/ (0.62)	0/276 (-)/ (0.57)	1931 (0.680/ 0.732)

†The number of utterances that the system cannot resolve the ambiguity of the domain because of Multiple domains that have the same selection score [(17 for baseline and 15 for proposed) (OK?)] method, respectively.

selected domain and the most probable domain for speech recognition results is incorrect. Furthermore, the precision for option (I) increased from 74% to 80%, which caused the F-measure to increase from 80% to 83%. While the recall of option (II) decreased from 75% to 62%, the precision of option (II) increased from 52% and 62% and the F-measure of option (II) remained at 61%. This demonstrates that the proposed method can detect option (III) without decreasing the accuracy achieved when detecting options (I) and (II).

However, the rate of domain selection errors was still 26.8%. Normally, the accuracy of option (II) relies heavily on the accuracy of the speech recognition. In this experiment, the accuracy of the speech recognition results was 63.3%, and it seems that this low accuracy caused the relatively high error rate of option (II). Additionally, there were 118 utterances that the proposed method incorrectly aligned with option (III). However, we believe such errors to be not so harmful because selecting option (III) normally leads to a system response to confirm the domain of a user's intention, and such confirmation never leads to incorrect domain changes, which are more harmful for users.

4.5 Summary

We summarize this chapter as follows.

- We demonstrated the advantages of the distributed architecture for multi-domain spoken dialogue systems considering domain extensibility and domain portability. The key component of the distributed architecture is our domain selection method. We pointed out that the domain selection procedure also has to satisfy the require-

ments of domain extensibility and domain portability.

- We proposed a new definition for the domain selection procedure, where the domain is selected from three options: the previously selected domain, the most probable domain from the speech recognition results, and other domains.

- We proposed using many dialogue features for robust domain selection. A decision tree constructed using these features demonstrated the importance of using the dialogue features. Experimental results showed that the proposed domain selection method was superior in terms of both accuracy and the ability to identify situations in which both the previously selected domain and the most probable domain for speech recognition results were incorrect.

Chapter 5

Vocabulary-Independent Indexing for Spoken Document Retrieval with Multi-stage Rescoring Method

5.1 Introduction

In this chapter, we focus on the subword-based method for the development of our open-vocabulary spoken term detection system. As mentioned in Chapter 2, conventional subword-based methods were normally slow and inaccurate. Some methods introduced the rescoring scheme to improve the accuracy, but the rescoring scheme made the subword-based methods further slow. In this chapter, we proposed the extended version of the conventional rescoring scheme in which three spoken term detection techniques are tandemly combined to narrow the search space in a stepwise fashion. We call this method the "mutli-stage rescoring". We proposed a very fast search method based on the voting and counting of the phoneme N-gram index for the first step of multi-stage rescoring method. We also proposed an acoustic rescoring method for the final step of the multi-stage rescoring method. The acoustic rescoring method is an accelerated version of the word spotting method, which made keyword detectoin significantly accurate. The proposed multi-stage rescoring method was much faster than conventional subword-based method and as accurate as a well-trained LVCSR-based method. In the next section, we describe the details of the proposed multi-stage rescoring method.

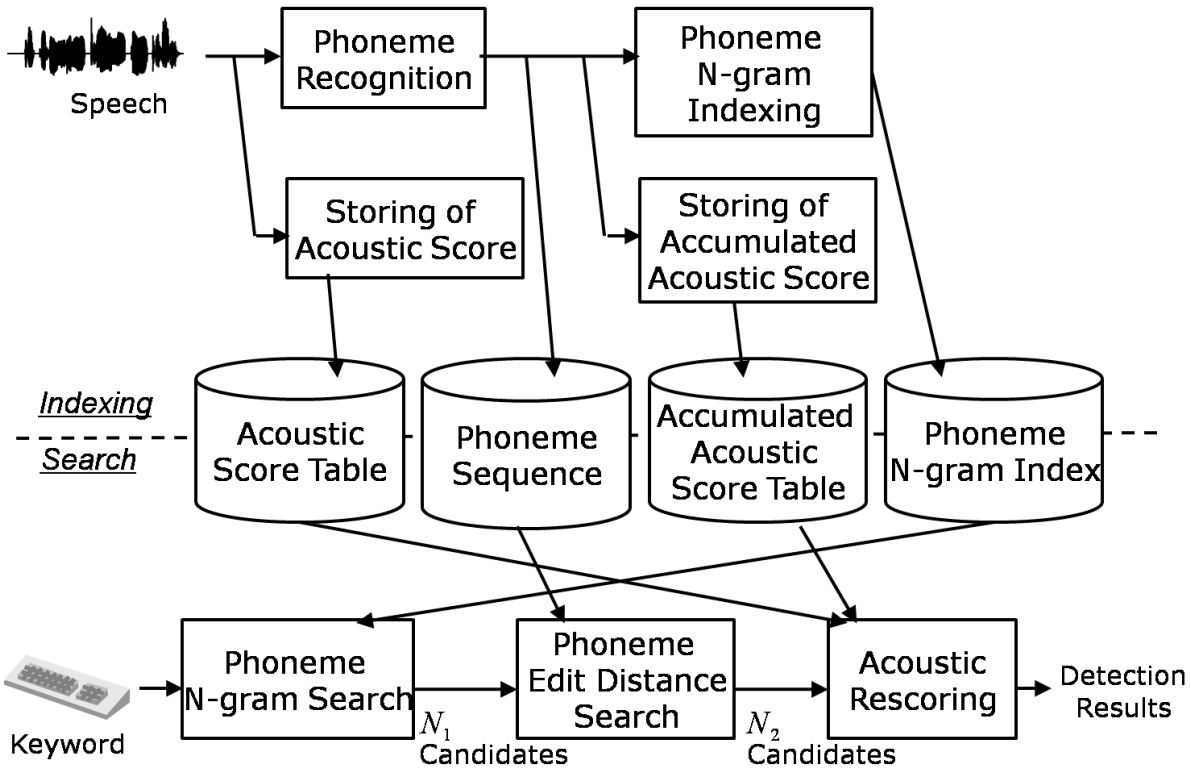


Figure 5.1: Overview of spoken term detection with multi-stage rescoring.

5.2 Open-Vocabulary Spoken Term Detection based on Multi-stage Rescoring

An overview of the proposed system is given in Figure 5.1. In the indexing module, four types of data, an acoustic score table, phoneme sequence, accumulated score table, and phoneme N-gram index, are produced. The phoneme N-gram index and the phoneme sequence are used for the phoneme N-gram search and the distance search modules, respectively. The acoustic and accumulated score tables are used for the acoustic rescoring method. The details of each module and data type are described in the next subsection.

5.2.1 Indexing Module

This subsection describes the details of the proposed indexing module.

1. Phoneme Recognition

   Features f_t are extracted for each time frame t. An acoustic score $P(f_t|S_i)$ for each state S_i is then calculated based on an acoustic model. In this study, phoneme

recognition based on a viterbi decoder is done using the acoustic score.

2. Acoustic score table
   An acoustic score table is a data structure that preserves the acoustic score $P(f_t|S_i)$ for the top-p highest states for each t. The acoustic score table can output $P(f_t|S_i)$ as a response for inputs of t and state number i[1]. Note that if there is no entry for t and i, the table outputs a small predefined value.

3. Accumulated score table
   An accumulated score table is a data structure that preserves accumulated phoneme recognition scores from time frame 0 to t according to the best hypothesis. The score is preserved for each t and includes the acoustic score, language model score, and insertion penalty.

4. Phoneme N-gram Index
   A continuing N phoneme sequence (phoneme N-gram) is extracted from the phoneme recognition results and stored with its recognized time. The phoneme N-gram index is a data structure to output the time regions in that a phoneme N-gram is observed by a response for an input of the phoneme N-gram.

5.2.2 Search Module

The lower part of Figure 5.1 indicates search modules. When a keyword is input, the phoneme N-gram search modules [40] first detect the candidate positions of the keyword. The top-N_1 candidates are then sent to the phoneme edit-distance search module and re-ranked by the module. Finally, the top-N_2 candidates are sent to the acoustic rescoring method based on the utterance verification techniques and re-ranked again. The three search methods above are confirmed to have different search speeds and accuracies; the earlier applied-one is faster but less accurate than the later one. Re-ranking the candidates in a stepwise fashion results in a good balance between search speed and search accuracy, which is evaluated in Section 5.3.

The details of each search module are given in the next subsection. Note that the acoustic rescoring method is one of the novel points of this study.

[1] In this study, we assume the tied-state triphone model, and that the state number indicates an index of a tied-state.

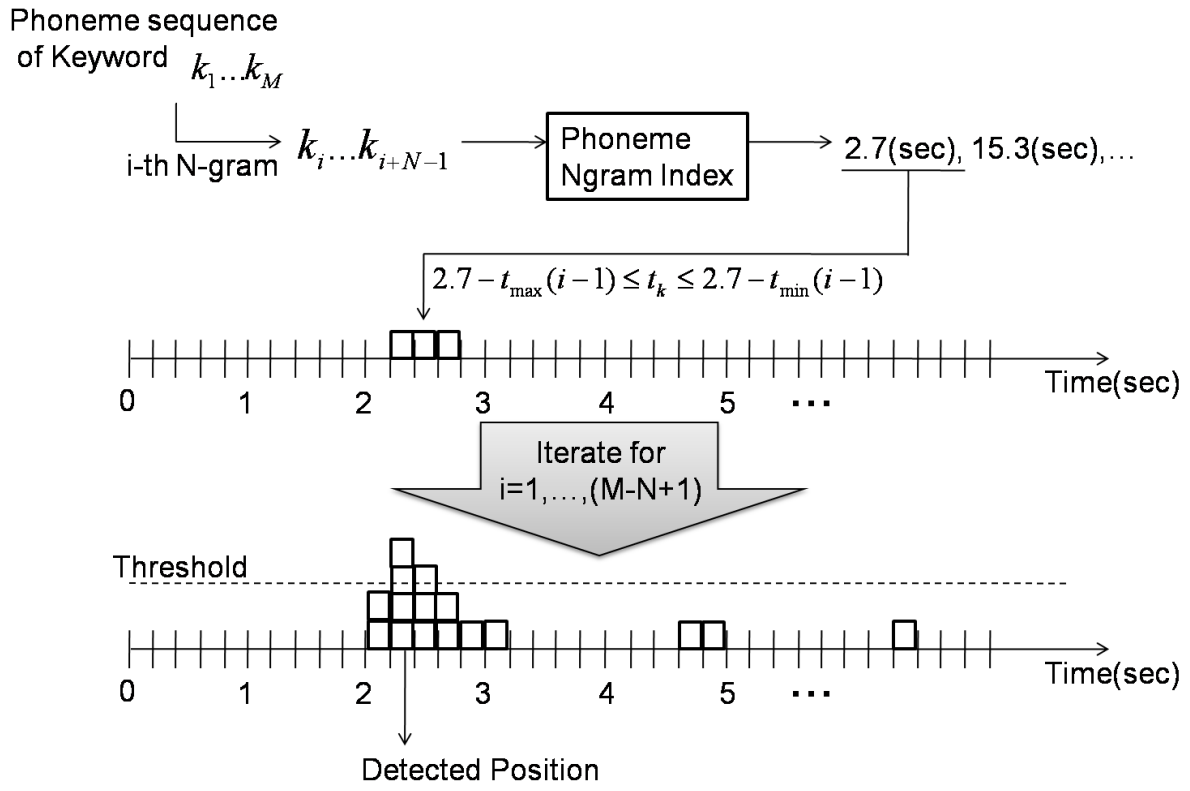


Figure 5.2: Spoken term detection using phoneme N-gram index.

Phoneme N-gram Search

In this search method, the phoneme N-gram index constructed in the indexing module is used to detect the keywords. We first prepare an array $S[T/l]$, where $T(sec)$ is the length of speech data and $l(sec)$ is the length of search bins. This array is initialized as 0 for each entry.

Consecutive N sequences $k_i...k_{i+N-1}(i = 1, .., (M - N + 1))$, which are called phoneme N-grams, are extracted from the phoneme expression of the keyword $k_1k_2...k_M$. For each phoneme N-gram, the time positions of the keyword in the speech data are then extracted by referring to the phoneme N-gram index.

Given an extracted t for a phoneme N-gram $k_i...k_{i+N-1}$, the beginning time of the keyword t_{k_1} can be estimated as $t_{k_1} = t - t_p * (i - 1)$, where t_p is the average duration per phoneme. Assuming $t_{min} < t_p < t_{max}$, t_{k_1} has the range of $t - t_{max} * (i - 1) \le t_{k_1} \le t - t_{min} * (i - 1)$. According to this assumption, we add 1 for the corresponding region of the array S for each extracted t for each $k_i...k_{i+N-1}$ in the phoneme expression of the

keyword.

Finally, a peak search by using the given threshold is executed for finding time positions where many phoneme N-grams are found. In this study, the following parameters were set according to preliminary experiments; $l = 0.2sec, t_{min} = 0.03sec, t_{max} = 0.12sec, and N = 3$.

Phoneme Matching based on Edit Distance

With this search method, the the keyword positions are detected by measuring the edit distance between the phoneme expression of the keyword and the recognized phoneme sequence. The edit distance is defined as the number of edit operations (substitution, insertion, and deletion) to convert the phoneme expression of the keyword into a form that can be included in the recognized phoneme sequence. In this study, we define the detection score as an inverse of the edit distance[2].

We used continuous dynamic programming for calculating the edit distance. As described in the beginning of this section, the N_1 candidate positions of a keyword are sent from the phoneme N-gram search module. The phoneme edit distance search module calculates the detection score by checking for each candidate position with some margin. We checked the utterances that included the candidate positions.

Acoustic Rescoring Method

The acoustic rescoring method is a search method based on the word spotting method [81] and modified for fast rescoring calculation.

We first describe the word spotting method that Kawahara et al. proposed [81]. With this method, the probability a keyword key in a region X, $P(*key*|X)$, is defined as follows ,

$$P(*key*|X) \simeq \frac{\max_{h_0 \in H_{syl,key,syl}} P(X|h_0)P(h_0)}{\max_{h_1 \in H_{syl}} P(X|h_1)P(h_1)} \tag{5.1}$$

Here, $H_{syl,key,syl}$ indicates the arbitrary sequence of states of HMMs that represent the syllable sequences including the keyword, and H_{syl} indicates the arbitrary sequence of the states of HMMs that represent the arbitrary syllable sequences.

[2]We used an inverse to ensure that the higher score indicates plausible results.

In Eq. (5.1) The denominator term indicates the likelihood of the best possible syllable sequence, and the numerator term indicates the likelihood of the best possible syllable sequence including the keyword. The calculation of the numerator term dominates almost all computational costs when calculating Eq. (5.1) for rescoring because the denominator term can be calculated in the phoneme recognition procedure in the indexing module.

We approximate Eq. (5.1) as follows.

$$P(*key*|X) \simeq \max_{Y \subseteq X} \frac{\max_{h_0 \in H_{key}} P(Y|h_0)P(h_0)}{\max_{h_1 \in H_{syl}} P(Y|h_1)P(h_1)} \tag{5.2}$$

We call this the acoustic rescoring method. Here, Y indicates an arbitrary region included in the observed region, and X H_{key} indicates an arbitrary sequence of states of HMMs that represent the keyword. Compared to Eq. (5.1), which takes into account the connections between the keyword and syllable sequences at the beginning and end of the keyword, Eq. (5.2) does not take into account such connections. Equation Eq. (5.2) produces higher value than Eq. (5.1) because of the lack of constraints on the connections; however, the differences in the score is expected to be limited because the differences will appear in the acoustic score of syllables at only the beginning and end of the keyword.

In Eq. (5.2), given t_1 and t_2 as the beginning and end points of region Y, we can calculate the denominator term as $R(t_2)/R(t_1)$, where $R(t)$ is the accumulated likelihood score at t that can be directly extracted from the accumulated score tables. Furthermore, given the state sequence h_0, we can calculate the numerator term of Eq. (5.2) as $P(Y|h_0)P(h_0) = \prod_{t=t_1}^{t_2} P(f_t|S_{h_0(t)})P(S_{h_0(t)}|S_{h_0(t-1)})$, where $P(f|S)$ is the acoustic score that can be extracted from the acoustic score table, and $P(S_{h_0(t)}|S_{h_0(t-1)})$ is the state transition probability from S_j to $S_i t$, which is included in the acoustic model. Here, $h_0(t)$ indicates h_0 at t. According to this information, we can solve Eq. (5.2) by using dynamic programming, which enables much faster computation of Eq. (5.2) compared to Eq. (5.1).

5.3 Experimental Evaluation

5.3.1 Evaluation Data and Measure

The accuracy of STD was evaluated experimentally. The settings for the experiment are mainly based on those reported by Itoh et al. [82], which was presented for the STD

task in the 9th NTCIR workshop. Spoken term detection experiment was conducted on speeches with a total duration of 604 hours from the Corpus of Spontaneous Japanese (CSJ) [83], which contains a total of 2,702 lecture recordings. We also used 39 hours of recordings extracted from CSJ for a mid-sized experiment. The 604-hours setting is called the "all set" and the 39-hours setting is called the "core set".

The evaluation was conducted as a two-fold cross validation. Lectures were first grouped into odd and even sets according to their file IDs. An acoustic model and a language model were trained using an odd (even) ID set and used to index the even (odd) ID set. Finally, indexes from the odd and even sets were combined and used for STD evaluation.

The keywords for the evaluation were prepared according to the study by Itoh et al. [82], which includes 100 known keywords (7.7 mora on average) for the all set, 50 known keywords (6.7 mora on average) and 50 unknown keywords (7.0 mora on average) for the core set. Itoh et al. [82] did not define the unknown keywords for the all set; therefore, we used "50 unknown keywords for the core set" even for the all set. Note that we used a different word unit from that used in [82], therefore, we modified the word boundary of the known keywords. We also used a different word dictionary compared to that used in [82]. As a result, some "unknown" keywords can consist of in-vocabulary words. For example, "チトー (Tito)" was defined as an unknown word; however, "チ (Ti)" and "トー (To)" are in the vocabulary and "チトー (Tito)" could consist of those words. To make such keywords unknown, we remove the utterances that include the unknown keyword when training the language models[3].

For the acoustic model of the baseline system, an ML-trained tied-state triphone HMM with 2,107 states was used. Each state was represented by eight mixtures of Gaussians. Thirteen MFCCs, including their delta coefficients and delta-delta coefficients with mean and variance normalization, were used.

A syllable 3-gram language model was used as the language model for the proposed method. Syllable recognition results were converted into phoneme sequences. We also used a word 3-gram language model for the word-based method as a baseline. Note that we eliminated utterances that include 50 unknown keywords for the core set when training both syllable language and word language models. For the word language model, we used

[3]In this study, we define unknown words as the words that not included in the training corpus (Cf. words that do not consist of in-vocabulary words).

Table 5.1: Phoneme recognition rates for syllable recognition system.

	Phoneme Correct Rate (%)	Phoneme Accuracy (%)
All set	85.2	78.5
Core set	86.8	79.9

Table 5.2: Word recognition rates for LVCSR.

	Word Correct Rate (%)	Word Accuracy (%)
All set	72.5	69.3
Core set	74.8	71.8

a short-unit defined for CSJ as a word unit and all words in the training corpus (except the eliminated utterances) were registered in the vocabulary. The vocabulary size was 95,278. For the words that only appear in odd (or even) ID data, we set a small 1-gram probability and back-off probability. For the syllable language model, we extended the syllable inventory to include long vowels (ex. "アー (a:)" or "カー (k a:)").

We used a 1-pass decoder based on a weighted finite state transducer for both word and syllable recognition. The word and phoneme recognition rates for the all set and core set are shown in Figure 5.1 and Figure 5.2[4]. Table 5.1 summarizes phoneme correctness and phoneme accuracy with the syllable 3-gram language model, and Table 5.2 summarizes word correctness and word accuracy with the word 3-gram language model. Note that we used slightly different settings that in [82]; therefore, word correctness and word accuracy had different maximum points; 1.9 and 0.1, respectively.

The detection results were evaluated on the basis of recall, precision, and F-measure (harmonic mean between recall and precision). We evaluated the correctness of the detection results by checking whether the utterance corresponding to the detected position includes the keyword. Recall and precision were averaged by keywords with an identical threshold. If no result was detected, the precision of the keywords was set to zero. In each evaluation, the thresholds were varied, and the best threshold that maximized the F-measure was selected.

[4]Language model weight and insertion penalty are set to 24 and 5 for word recognition and 10 and 0 for syllable recognition. The details of these settings are discussed in Section 5.3.2 and Section 5.3.2.

Table 5.3: F-measure of acoustic rescoring and size of acoustic score table.

p	Size of Acoustic Score Table (GB) Core Set / All Set	F-measure of Acoustic Rescoring Core Set / All Set
10	0.32 / 4.92	47.8 / 43.7
20	0.63 / 9.84	64.9 / 56.1
30	0.95 / 14.8	68.9 / 60.9
40	1.27 / 19.7	70.9 / 63.2
50	1.58 / 24.6	71.3 / 63.6
60	1.90 / 29.5	71.5 / 64.1

5.3.2 Parameter Settings

Parameters for Acoustic Rescoring Method

The size of the acoustic score table and F-measure for the acoustic rescoring method with several parameters ps are shown in Figure 5.3[5].

In this phase, we used 100 keywords (50 known and 50 unknown keywords) for the core set, and 150 keywords (100 known and 50 unknown keywords) for the all set. Note that the acoustic rescoring method was applied for 10,000 candidates output using the phoneme N-gram search method.

From Figure 5.3, we confirmed that the size of the acoustic score table increased according to p. We also confirmed that a larger p normally produces a better F-measure, but the improvement nearly converged at about $p = 50$. From the viewpoint of system design, the size of the acoustic table should be small, and F-measure should be high, which indicates that there is a trade-off between the size of the acoustic table and F-measure. We set $p = 50$ for the following experiments.

Parameter for Large-Vocabulary Continuous Speech Recognition

We investigated the influence of the language model weight and insertion penalty from the perspectives of word accuracy and search accuracy. We used the core set and 50 known keywords. The word accuracy and search accuracy (F-measure) are given in Figure 5.3 and Figure 5.4, respectively. While the word accuracy was highest when the language

[5]In this study, we used 8-bit quantization for the acoustic score table for saving storage space. In quantization procedure, the acoustic score was divided by the maximum acoustic score observed on the frame to normalize the range of scores. This normalization was also applied for the accumulated score table so as not to affect 5.2.

Table 5.4: Word accuracy and search accuracy of word-based method.

	Word Accuracy (%)	Recall (%)	Precision (%)	F-measure (%)
Word(RecogOPT)	**71.8**	59.4	81.9	68.9
Word(SearchOPT)	70.3	59.6	93.1	**72.7**

model weight was 24 and the insertion penalty was 5, search accuracy was highest when the language model weight was 16 and the insertion penalty was 10.

The recall, precision, and F-measure for those two settings are listed in Table 5.4. In this table, Word(RecogOPT) indicates the values at the settings that maximize word accuracy, and Word(SearchOPT) indicates the values at the settings that maximize search accuracy. We could confirm that while the precision at Word(RecogOPT) was 81.9%, that at Word(searchOPT) was 93.1%, and this is the main difference between the two settings. Specifically, while there were two keywords that were not detected only one candidate at Word(searchOPT), there were seven such keywords at Word(RecogOPT). We defined the precisions for such keywords as 0; therefore, those keywords degrade the total precision. For example, the keyword "エベレスト街道 (Everest kaido)" was correctly recognized at the setting of Word(searchOPT), but incorrectly recognized at the setting of Word(RecogOPT) as "ですとか移動 (desu to ka ido)". Considering the fact that we cannot determine the keyword set in advance, we normally cannot use the setting of Word(searchOPT). Therefore, we used both settings for the following experiments.

Parameters for Syllable Recognition

We investigated the phoneme accuracy of the syllable recognition results and F-measure of the phoneme matching method based on the recognition results by changing the language model weight and insertion penalty. In this experiment, we used the core set with 50 known and 50 unknown keywords. The phoneme accuracy is shown in Figure 5.5. The F-measures for the 50 known and 50 unknown keywords are shown in Figure 5.6 and Figure 5.7, respectively.

The phoneme accuracy and search accuracy for known keywords was better when the language model weight was larger. In contrast, the correlation between the language model weight and search accuracy for unknown keywords was not observed, and the best search accuracy was obtained when the language model weight was 10. We also found

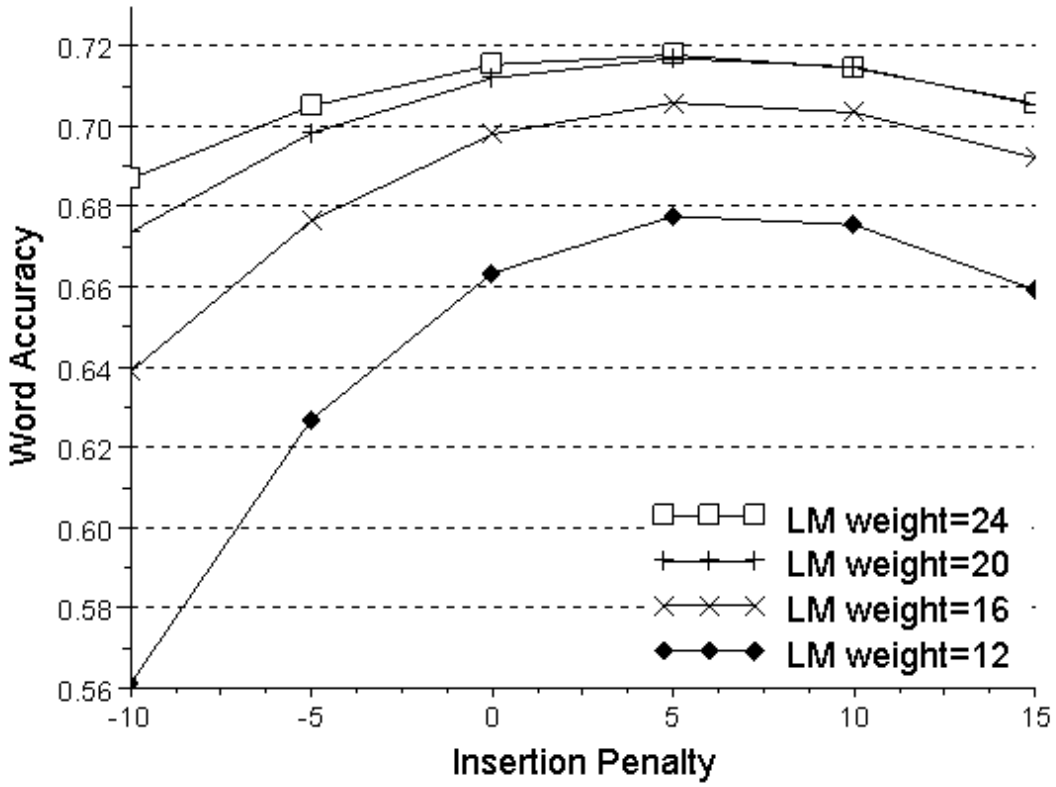


Figure 5.3: Word accuracy for core set.

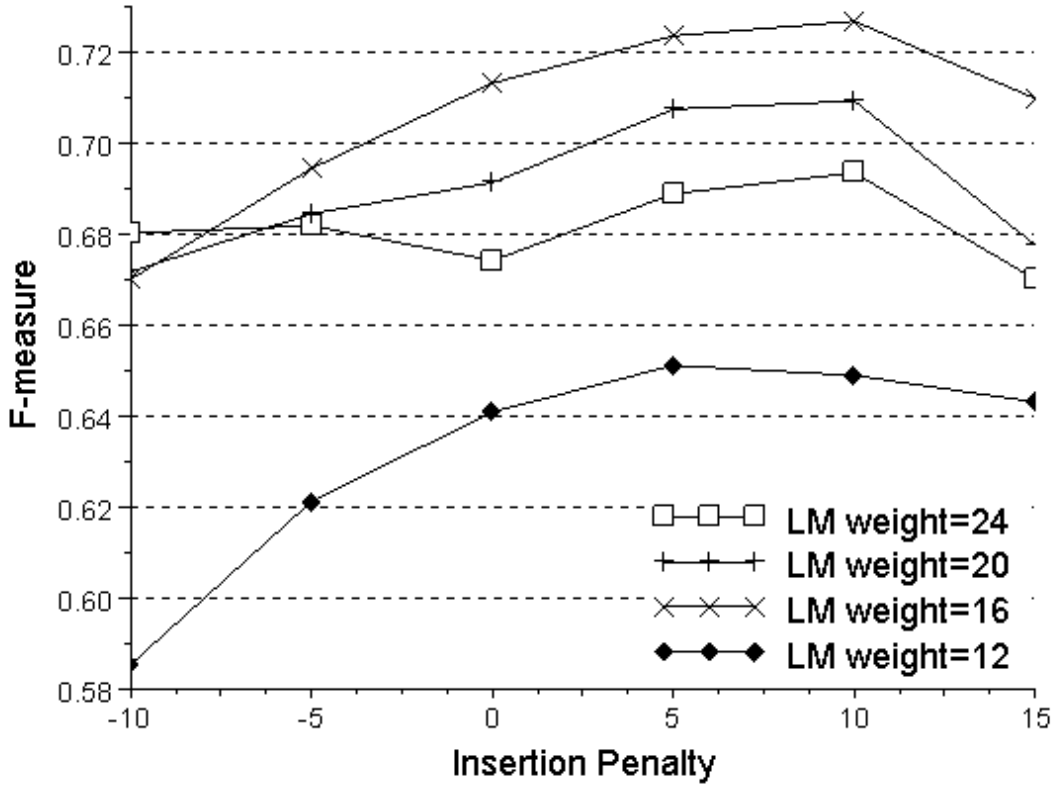


Figure 5.4: Search accuracy of word-based method for core set (50 known keywords).

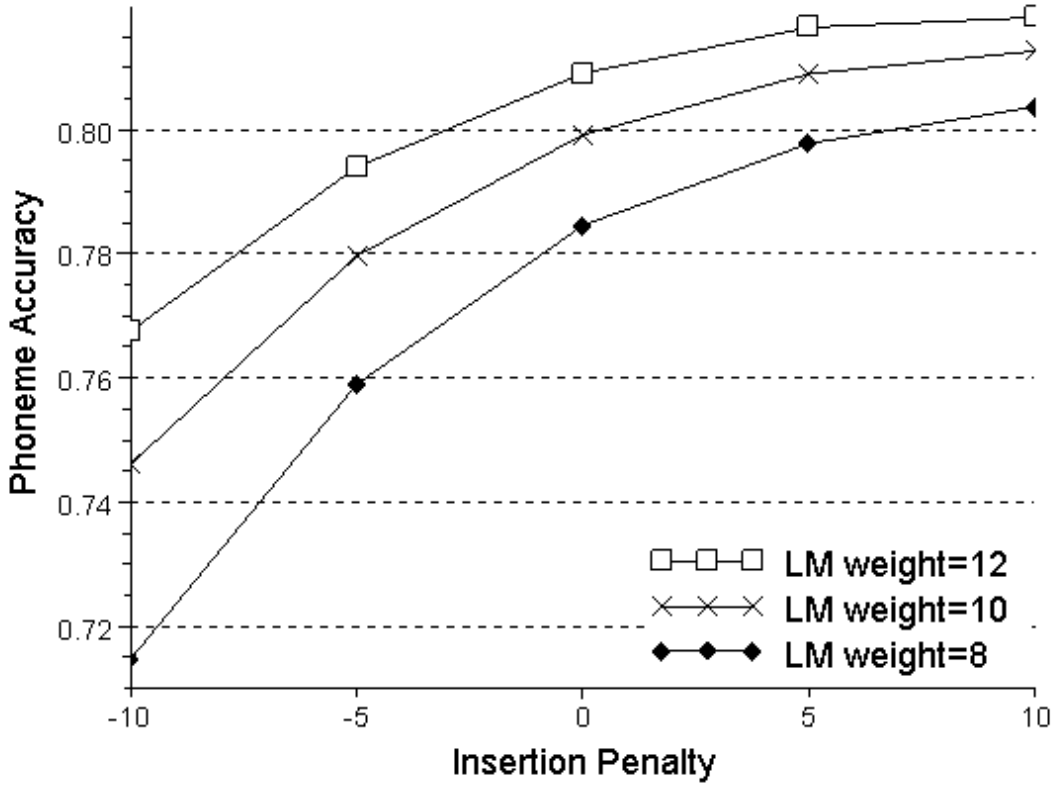


Figure 5.5: Phoneme accuracy for core set.

that while the phoneme recognition was good when the insertion penalty was large, the search accuracy for both keyword sets was good when the insertion penalty was small if the insertion penalty was from -10 to 10. From these observations, we took the balanced setting in which the language model weight and the insertion penalty were set to 10 and 0, respectively. It should be noted that we used the same parameters for the proposed acoustic rescoring method. It should also be noted that the F-measure for the unknown keywords was 59.6% in [82] with similar settings, which is 9.4 points higher than our result. We believe this difference is caused by the difference in the acoustic and language models[6]. The results in [82] are cited when comparison is needed in the remaining sections.

[6]For example, while the long vowel [82] was represented by a double vowel, we expanded syllable inventories to contain syllables with long vowels (ex. "fu u ri e" ([82]) and "fu: ri e" (our expression) for the keywords "フーリエ (Fourier)"). Our syllables are on average longer than those defined in [82], which may make our language model stronger for known keywords and weaker for unknown keywords.

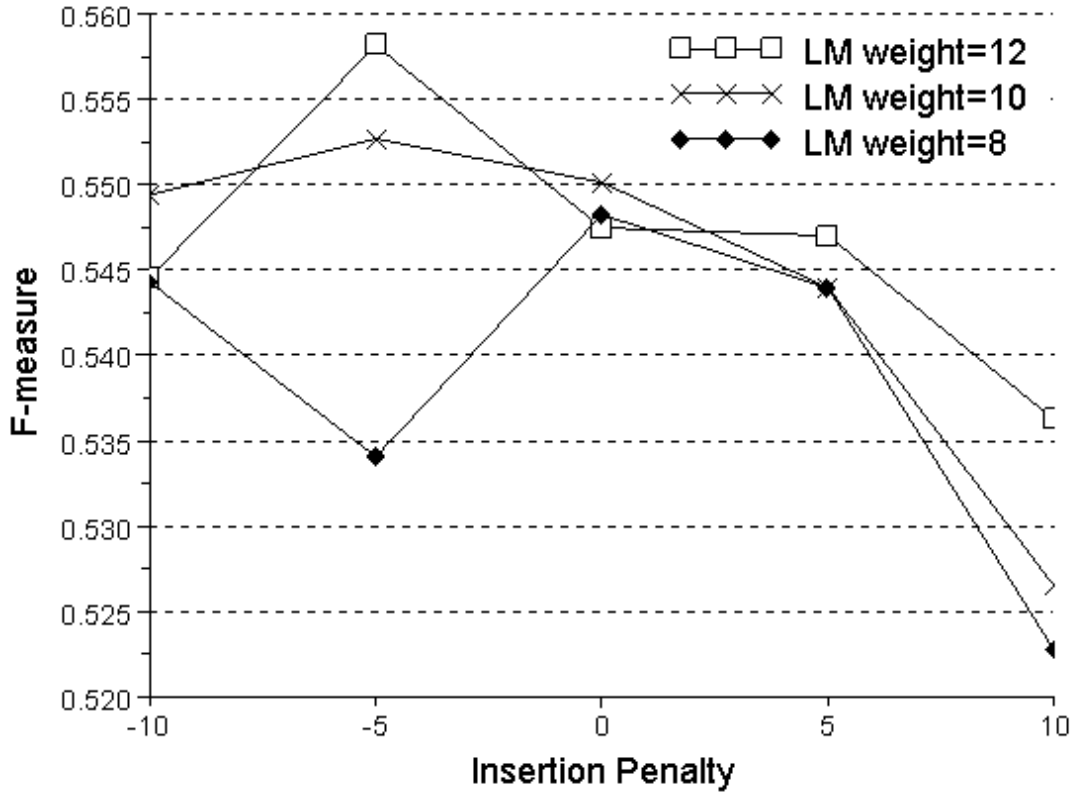


Figure 5.6: Search accuracy of edit-distance-based method for core set (50 known keywords).

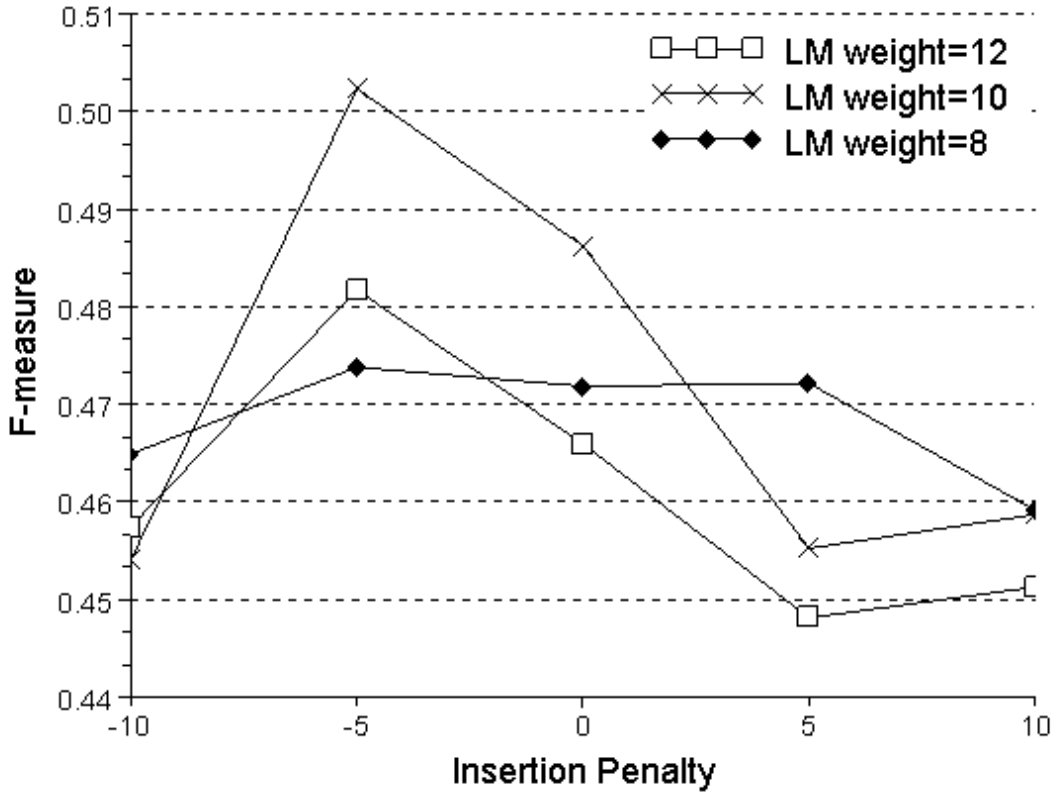


Figure 5.7: Search accuracy of edit-distance-based method for core set (50 unknown keywords).

Table 5.5: Index size for all set.

Index Type	Index Size
Phoneme sequence	132 MB
Phoneme N-gram index	130 MB
Acoustic score table	24.6 GB
Accumulated acoustic score table	1.02 GB

5.3.3 Index Size and Processing Time of Indexing

In the proposed indexing procedure, four types of data – phoneme N-gram index, phoneme sequence, accumulated score table and acoustic score table – are created. The sizes for each data type created for the all set are listed in Table 5.5. Parameter p of the acoustic score table was set to 50. The acoustic and the accumulated score tables are much larger than other two types of data. In this study, the phoneme N-gram index and phoneme sequence were stored on memory, and the accumulated and acoustic score tables were stored on a solid-state disk (SSD).

We used a Linux machine with Xeon X3430 (2.40 GHz) and 8-GB memory to measure the processing time. Each calculation was done using a single core. Because we took into account the accessing time to the SSD when measuring the total processing time, we cleared the page cache of the machine each time when measuring the processing time. We used the Crucial Real SSD 128-GB model.

5.3.4 Evaluation Results

Evaluation of Multi-stage Rescoring Method

We first investigated the search speed and accuracy for the individual search methods used in the proposed multi-stage rescoring method. Figure 5.6 indicates the results for the all set with 100 known keywords. In this table, search speed indicates the duration of speech the search method can process within one second. The proposed acoustic rescoring method was applied for 10,000-candidate output by using the phoneme N-gram search method. From this table, we could confirm there was a trade-off between search speed and search accuracy. We also confirmed that the proposed acoustic rescoring method was more accurate than the phoneme matching method based on edit distance.

We then evaluated the search accuracy (F-measure) and search speed of the proposed multi-stage rescoring method for the all set with several settings of candidates N_1, N_2. Ta-

Table 5.6: Search speed and accuracy for all set with 100 known keywords.

	Search Speed (hour/sec)	F-measure (%)
Phoneme N-gram	**4066**	37.8
Edit distance	173	52.0
Acoustic rescoring	0.31	**64.0**

Table 5.7: Impact of N_1on multi-stage rescoring (all set with 100 known keywords).

N_1	N_2	Total Processing Time (sec) Full/Edit Distance	F-measure (%)
10000	500	2.53 / 0.11	64.7
7500	500	2.59 / 0.083	65.0
5000	500	2.55 / 0.056	64.4
2500	500	2.58 / 0.029	63.6
1000	500	2.58 / 0.012	61.0
500	500	2.53 / 0.0060	59.9

ble 5.7 lists the results when $N_2 = 500$ and with varying N_1. The total search time (Full) and time for phoneme matching based on edit distance (Edit Distance) are shown. At $N_1 = 10000$ and $N_2 = 500$, it took 0.25 sec, 0.11 sec, and 2.17 sec for the phoneme N-gram search, phoneme matching based on edit distance, and acoustic rescoring method, respectively. Note that phoneme matching was so fast that the number of candidates N_1 had a small impact on total search time. We confirmed that search accuracy slightly improves as N_1 increases. We also confirmed that search accuracy converged with slight fluctuation. These observations indicate the importance of phoneme matching for rescoring the results.

Note that the F-measure of the multi-stage rescoring method with N_1 over 5000 was 0.4 to 1.0 points higher than that of the single acoustic rescoring method in Table 5.6. This may be because the tandem combination of the three methods had an effect of filtering incorrect results, which may improve search accuracy.

Next, the results when N_1 was set to 5000 and with varying N_2 are shown in Figure 5.8. The total search time (Full) and time for the acoustic rescoring method (Acoustic Rescoring) are shown. We can see that a larger N_2 resulted in more accurate search accuracy, and this improvement saturated at about $N_2 = 250$. We also found that the total search time was mainly dependent on N_2.

The results in Table 5.7 and Table 5.8, and the results of the phoneme N-gram method

Table 5.8: Impact of N_2 on multi-stage rescoring (all set with 100 known keywords).

N_1	N_2	Processing Time (sec) Full/Acoustic Rescoring	F-measure (%)
5000	500	2.55 / 2.24	64.4
5000	250	1.42 / 1.12	64.4
5000	100	0.70 / 0.43	62.5
5000	50	0.44 / 0.20	58.7
5000	0	0.17 / 0.00	50.2

Table 5.9: Comparison of indexing speed.

	Recognition Speed (xRT)
With syllable recognition	0.11
With word recognition	0.35

and phoneme matching are plotted in Figure 5.8. The horizontal axis indicates the average search time for the all set, and the vertical axis indicates the F-measure. The proposed multi-stage rescoring method resolved the trade-off between search speed and search accuracy, and obtained highly accurate results with little search time.

Indexing Speed for Each Method

Indexing speeds for LVCSR are listed in Table 5.9. When calculating the indexing speed, we included not only the time for speech recognition but also that for creating the index data. The large vocabulary continuouse speech recognition (LVCSR) was evaluated at a setting that maximized word accuracy. The beam width was set to achieve enough accuracy. As shown in Table 5.9, syllable recognition was faster than LVCSR. This is because the syllable recognition module search for recognition results from a hypothesis space consisting of about 250 syllable inventories[7], the LVCSR module required searching for recognition results from a much larger hypothesis space consisting of about one hundred thousand words. From these results, we concluded that the proposed method with syllable recognition can make indexes faster than that with LVCSR, which is important when making an index for a large-scale speech database.

[7]We extended the syllable inventory to include long vowels, as previously mentioned.

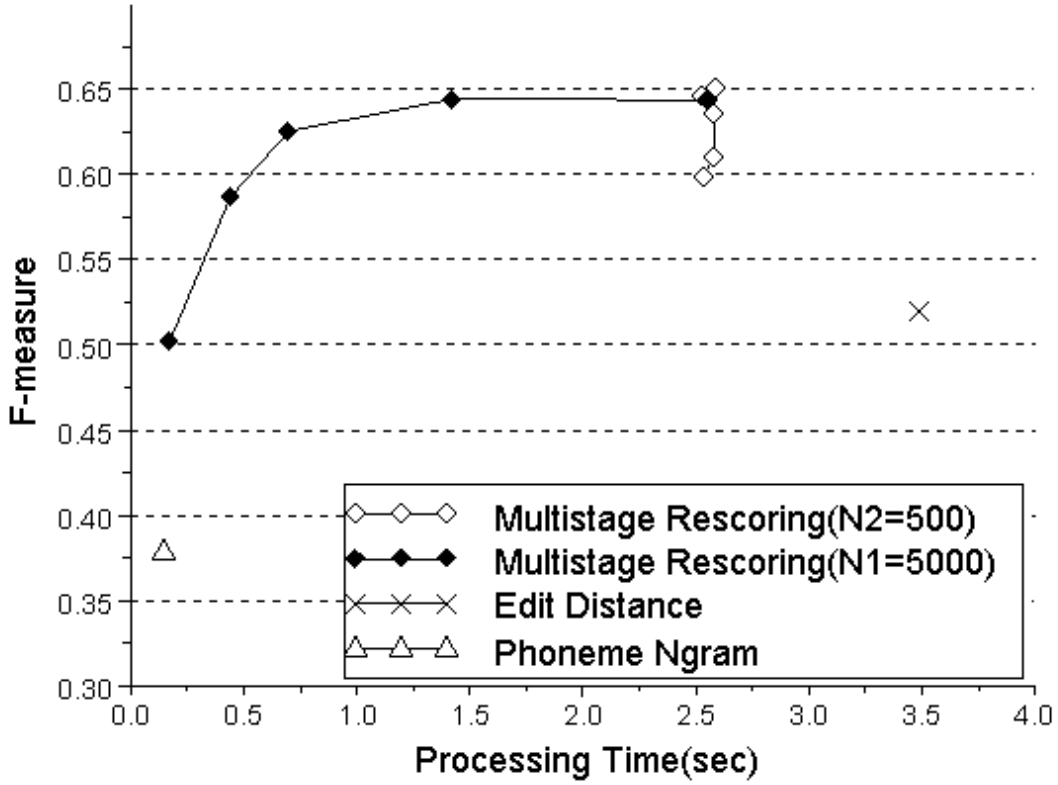


Figure 5.8: Search accuracy and processing time for all set.

Search Accuracy for Known Keywords

This section describes the investigation results of search accuracy for known keywords with the proposed multi-stage rescoring methods. We used the all set and core set. In a test set, 100 known keywords for the all set appeared 53.1 times on average, and 50 known keywords for the core set appeared 14.5 times on average. We set the parameters as $N_1 = 5000 and N_2 = 250$.

Table 5.10 and Table 5.11 list the results. According to [82], we tested a method based on strict word matching based on the LVCSR results (Word) and a phoneme matching method based on the edit distance with the syllable recognition results (Edit Distance). For the former, we tested RecogOpt and SearchOpt, which were described in the previous section.

From Table 5.10, in the core set, we first found that the edit distance was worse in

Table 5.10: Search accuracy for core set (50 known keywords).

	Recall (%)	Precision (%)	F-measure (%)
Word(RecogOpt)	59.4	81.9	68.9
Word(SearchOpt)	59.6	93.1	72.7
Edit distance	53.4	56.7	55.0
Proposed [Which one?]	66.3	70.9	68.5

every metric (recall, precision and F-measure). On the contrary, strict word matching was highly accurate on both settings.

Compared to these results, the multi-stage rescoring method (Proposed) obtained a 13.5-point higher F-measure than with the phoneme matching method, and achieved similar accuracy to the word matching method. The highest F-measure 72.7% was obtained when using the word matching method with optimal recognition settings for searching keywords. However, considering that the settings were optimized with a given keyword set, the accuracy of the word matching method was from 68.9% to 72.7%. Therefore, we believe that the search accuracy of the proposed multi-stage rescoring method (68.5% F-measure) was almost the same as that of the word matching method.

The results for the all set was the same as those of the core set 5.11, however, the F-measure of the proposed method was 3.5 to 4.0 points lower than that with the word matching method. While the word matching method at Word(RecogOpt) produced 4.2 false alarms on average, the proposed method produced 9.5 false alarms on average, which was the main difference in the two methods. Note that the keyword in the core set was uttered once every 2.7 hours on average, the keyword in the all set was uttered once every 11.4 hours; therefore, the all set had 4.2 times more probably to produce false alarms. Regarding precision, the proposed method was worse than the word matching method, which means the proposed method is likely to produce false detections, which we believe was a cause of lower F-measure in the evaluation with the all set[8].

Search Accuracy for Unknown Keywords

This section describes the investigation results of search accuracy for unknown keywords with the proposed methods. We used both the all set and core set. In a test set, 50 unknown keywords appeared on average 4.6 times and 5.7 times for the core set and all

[8]Note that the F-measure of the word matching method was 63.5% in [82], which was 0.9 points lower than that with the proposed method.

Table 5.11: Search accuracy for all set (100 known keywords).

	Recall (%)	Precision (%)	F-measure (%)
Word(RecogOpt)	58.2	81.5	67.9
Word(SearchOpt)	56.7	86.3	68.4
Edit distance	46.7	58.7	52.0
Proposed	55.6	76.5	64.4

set, respectively. We set the parameters to $N_1 = 5000, N_2 = 250$. Note that, as described in Section 5.3.1, we defined unknown keywords as words not included in the training transcription (cf. out-of-dictionary). As a result, 17 keywords consisted of dictionary words, and they could be detected even with the word matching method.

Table 5.12 and Table 5.13 lists the search accuracies for the core set and all set, respectively. We confirmed that the F-measure of the word matching method was only about 5% for both sets. Note that even for very low F-measure and low precision, the detected results were all correct. Our evaluation framework in this chapter defined that the precision of the keyword in which the system could detect any result was set to 0%, and such keywords degraded precision.

The phoneme matching method largely improved accuracy, resulting in an F-measure of 48.6% and 39.1% for the core set and all set, respectively.

The proposed method further improved accuracy, resulting in an F-measure of 78.1% and 67.8% for the core set and all set, respectively. Note that Itoh et al. [82] reported an F-measure of 59.6% for the core set by using the phoneme matching method, which is better than our results (48.6%). However, the results of the proposed method (78.1%) still showed sufficient improvement.

Compared to the results with known keywords (Table 5.10 and Table 5.11), higher results were obtained from the proposed method. Considering that the syllable language model used in the proposed method would be quite affected by whether the keyword was known or unknown, we believed that inheritance difficulty of the keywords would be reflected in those results. Specifically, while the known keywords appeared 53.1 times on average in the all set, the unknown keywords appeared only 5.7 times on average. As a result, with the setting $N_2 = 250$, while about five candidates per correct result were rescored with the acoustic rescoring method for known keywords, about 40 candidates per correct result can be rescored for unknown keywords, which could improve recall of the

Table 5.12: Search accuracy for core set (50 unknown keywords).

	Recall (%)	Precision (%)	F-measure (%)
Word(RecogOpt)	3.4	6.0	4.3
Word(SearchOpt)	4.0	10.0	5.7
Edit distance	59.8	41.0	48.6
Proposed	73.3	83.6	78.1

Table 5.13: Search accuracy for all set (50 unknown keywords).

	Recall (%)	Precision (%)	F-measure (%)
Word(RecogOpt)	3.4	8.0	4.8
Word(SearchOpt)	4.7	12.0	6.8
Edit distance	45.7	34.2	39.1
Proposed	66.7	68.9	67.8

results. The proposed method produced accurate results without taking into account if the keyword was known or unknown. Such properties were not observed with conventional methods, and we believe it is important for developing open-ended spoken document retrieval systems.

5.4 Summary

In this chapter, we focused on spoken term detection systems and proposed a fast and accurate open vocabulary indexing method, which can detect arbitrary keywords. The results are summarized as follows.

- We proposed a spoken term detection system in which three spoken term detection techniques are tandemly combined to narrow the search space in a stepwise fashion. We proposed a very fast search method based on the voting and counting of the phoneme N-gram index for the first step of multi-stage rescoring method. We also proposed an acoustic rescoring method for the final step of the multi-stage rescoring method. The acoustic rescoring method is an accelerated version of the word spotting method, which made keyword detectoin significantly accurate. These methods enabled fast and accurate open vocabulary detection.

- The experimental results showed that the proposed method can detect an unknown keyword that occurred only 5.7 times on average in 604 hours of lecture recordings

within 1.7 seconds with an F-measure of 67.8%. The accuracy of the proposed multi-stage rescoring method was much better than the conventional open vocabulary methods, and just 3.5-4.0 points worse than the well-tuned LVCSR-based method, which cannot detect unknown keywords.

Chapter 6

Robust and Compact Index Combination based on Index Selection with Out-of-Vocabulary Region Estimator

6.1 Introduction

In Chapter 5, we described a fast and accurate spoken term detection system that can search for arbitrary keywords without the restriction of a dictionary. With unknown keywords, the proposed system achieved much more accurate results than conventional methods, but with known keywords, it was worse than the word-based conventional method, especially when there was a very large amount of speech data. This indicates that there is room for further improvement of search accuracy by combining word-based systems that could provide domain-specific and vocabulary-specific knowledge to the search system. Therefore, in this chapter, we investigate system combination methods that combine various systems to obtain more accurate results.

A simple way to use word-based and subword-based systems is switching between the two systems depending on whether the keyword is known or unknown. This method is simple, but its search accuracy never outperforms that of the original systems. Recently, there have been proposals of methods combining multiple types of indices that achieve high detection accuracy for both known and unknown keywords [45–47]. For example, the best performance in the latest NTCIR STD evaluation [26] was obtained by a method that combines the outputs of ten different recognizers [47]. There are many variations of index-combination methods: subword unit type (word/phoneme [32,33], original subwords [48]), index format (lattice [32], confusion network [33, 46, 47]), and score calculation (modified

edit-distance [47], weighted-sum [46]).

One disadvantage of the multiple index-combination method is its large index size. Obviously, as many indices are combined, the index size becomes larger, and a larger index not only increases storage costs but also slows the search speed [26,47]. A confidence measure (like the one in [49]) could be used to identify the redundant portions of an index made from a single recognizer [50–52], but it is not always easy to extend this method to a combined index made from multiple recognizers because confidence measures from different recognizers are often biased differently. Furthermore, the confidence measure of a region that contains OOVs tends to have a small value, and therefore confidence-measure-based index pruning may degrade the accuracy of OOV queries.

In this chapter, we propose a novel index combination method for spoken term detection. In our method, outputs from four different recognizers—word, syllable, word-syllable, and fragment—are combined into one confusion network. A novel index-selection method for the multiple index-combination method is then used to suppress the increase of the index size. Experimental results using 39 hours of Japanese lecture recordings showed that the index-selection method could reduce the index size of the best confusion network by 22.7% while maintaining high accuracy. Compared with the best phoneme index from a single recognizer, the proposed method achieved 16.3% and 16.0% relative error reductions for IV and OOV queries, respectively, without increasing index size.

There have been many studies that propose combining a word index and a subword index [31–33]. For such indices, it is obvious that the subword-index is redundant for regions in which words have already been correctly recognized. Other studies have explored pruning the subword index in accordance with IV-word existence score (e.g., word posteriori probability) [33,34]. Our proposed index-selection method can be considered an extension of the above works to a state-of-the art multiple index combination method [45–47] for which an obvious index-selection method does not currently exist.

6.2 Spoken Term Detection System with Multiple Indices

Figure 6.1 depicts the system overview. The spoken term detection system consists of two modules: an indexing module and a search module. The indexing module starts operation when new speech data is added to the system and makes an index optimized

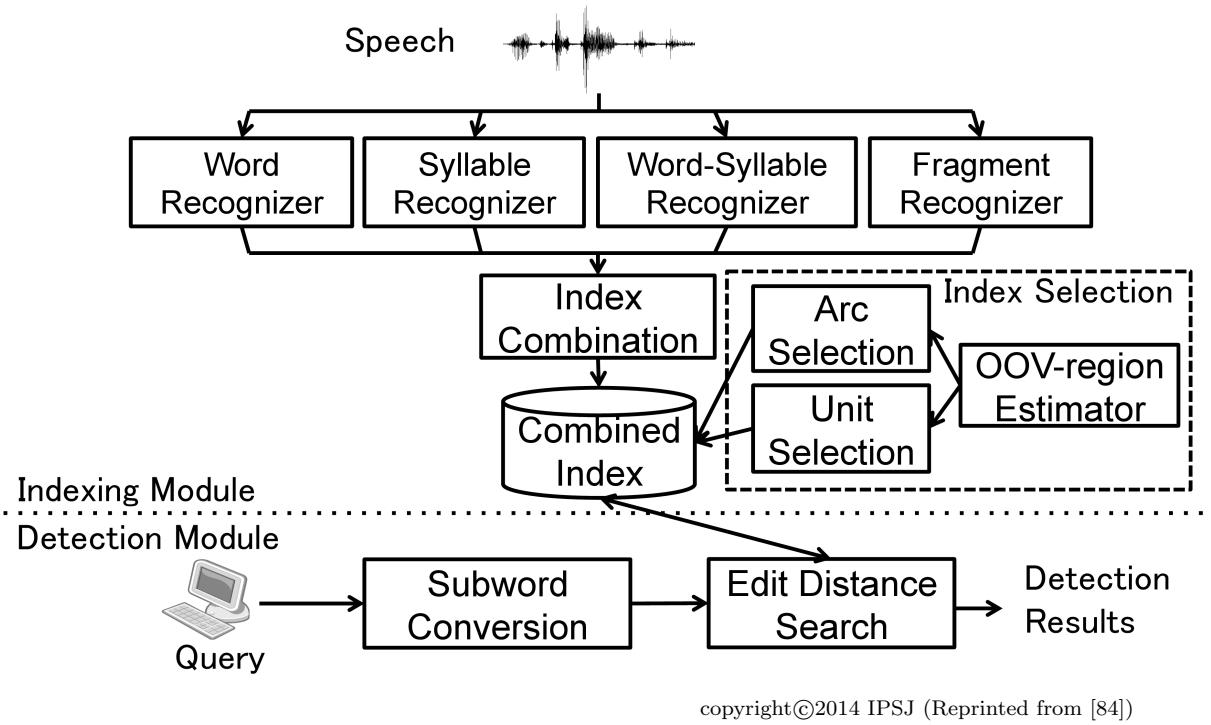


Figure 6.1: Overview of spoken term detection system.

for spoken term detection. The search module starts operation when a user inputs a query into the system and detects the positions at which the keyword was uttered in the speech database.

In the indexing module, we used four types of speech recognizers, all containing different language models, listed below.

Word: Word language model trained from a text corpus.

Syllable: Syllable language model trained from a corpus in which all content is converted into syllables.

Word-Syllable: Word and syllable mixed language model trained from a corpus in which only rare words (occurring less than 2 times) are converted into syllables.

Fragment: Language model trained from a fragment corpus. To make this corpus, we first prepare a syllable corpus the same as that used for the syllable language model. We then iteratively join two symbols that mostly occur successively in the corpus. The iteration is stopped when the average length of the joined syllables, which we call a "fragment", becomes the same as the average word length. This model can be regarded as a variation of the model proposed in [85].

Recognition results are then converted into subword sequences and combined into one

Recognizer	Recognition Result
Word	o N g a k u o t a m e n i
Syllable	a N n a k a t a n a i n i
Word-Syllable	o N g a k u w a t a n e n i
Fragment	o N g a k u z a N n e N n i

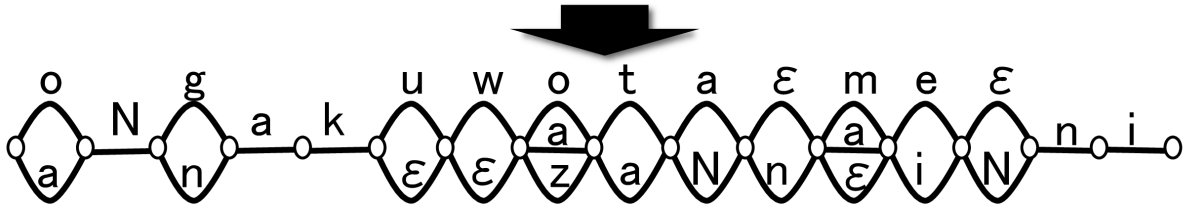


Figure 6.2: Index combination as a confusion network.

index. In this study, **syllables** or **phonemes** are used as the subword unit. We use the transition-network combination method that Nishizaki et al. proposed [47], where the best path of a word recognizer's result is regarded as a reference sequence and the other recognizers' results are aligned with the reference sequence so as to minimize the edit distance between them. Aligned sequences can be seen as a kind of confusion network, originally called a "transition network (TN)" in [47]. In this study, we call the network developed by the above protocol a "transition network" and the network developed by the method described in [86] a "confusion network". Figure 6.2 shows an example of phoneme-based index combination. Results from multiple recognizers are first aligned with each other and then combined into a transition network. Arcs with a ε mark in the transition network indicate epsilon transition arcs.

In the search module, an input query is first converted into a subword sequence (query subword) according to pronunciation rules. Then, an edit distance E, which is the cost of manipulation to make the query subword included in TN, is calculated. Finally, a detection score S is calculated as E normalized by the number of subwords in the query subword N_q, as

$$S = 1 - \frac{E}{N_q} \tag{6.1}$$

The search module calculates detection scores for each utterance and outputs the detected results according to the detection score in descending order.

Edit distance E can be calculated efficiently by using a dynamic programming method. Given $Arc(i)$ ($i = \{1, ..., I\}$) as the i-th arcs in a transition network, $a \in Arc(i)$ as non-null elements of $Arc(i)$, and q_j ($j = \{1, ..., N_q\}$) as j-th subword in the query subword. At that time, edit distance E between the transition network and the query subword can be calculated by the following procedures.

Procedure 1: Initialization for $i = \{0, ..., I\}, j = \{0, ..., N_q\}$.

$$D(i,j) = \begin{cases} 0 & if\ j = 0 \\ N_q & otherwise \end{cases} \tag{6.2}$$

Procedure 2: Iteration for $i = \{1, ..., I\}, j = \{1, ..., N_q\}$.

$$D(i,j) = \min \begin{cases} \min\limits_{a \in Arc(i)} \{D(i-1, j-1) + sub(q_j, a)\} \\ D(i-1, j) + ins(Arc(i)) \\ D(i, j-1) + del \end{cases} \tag{6.3}$$

Procedure 3: Minimum calculation.

$$E = \min_{i \in \{1,...,I\}} D(i, N_q) \tag{6.4}$$

Here, $sub(q_j, a)$ is a substitution cost of subword q_j and the input symbol of arc a. In this work, the substitution cost was set to 0 if the input symbol of a is the same as q_j, and otherwise the cost was set to 1[1]. $ins(Arc(i))$ is an insertion cost for inserting the input symbol of an arc to the query subword. The insertion cost was set to 0 if $Arc(i)$ contains a null arc ϵ and set to 1 otherwise. Finally, del is a deletion cost for deleting a subword in the query subword, and in this work the cost was set to 1. The computational cost of the procedure is mainly dependent on the iteration in procedure 2. Given the number of arcs M in the index, the first and second columns of (6.3) require $N_q \cdot M$ operation, the third column requires $N_q \cdot I$ operation, and finally $N_q \cdot M$ operation is required for a minimum calculation. Normally, the operation for the first and second columns and the minimum calculation limit the operational speed to $O(N_q \cdot M)$.

6.3 Index Selection based on OOV-Region Estimator

6.3.1 Strategy for Index Selection

The aim of the index selection is to reduce index size without degrading the accuracy of the multiple index-combination method. We introduce the OOV-region estimator to

[1]In this work, we did not use the "Voting" or "ArcWidth" techniques proposed in [47].

select redundant indices. The OOV-region estimator is a technique that estimates the existence probability of OOVs in an observed region [87–89]. Note that the classifier does not care which OOV-term exists[2]. We reduce an index with two methods: arc selection and unit selection.

The arc selection method removes index arcs originating from a specific recognizer if the OOV-region estimator score of the arcs is smaller than (or greater than) the threshold. The assumption behind this method is that some recognizers' outputs will contribute to detecting OOV (or IV) queries but others will not make any contribution to IV (or OOV) query detection. In our experiments, we removed arcs if either of the following conditions, both of which were defined by preliminary experiments, were true.

1. The arc originated from the word recognizer only, and the OOV-region estimator score of the region belonged to the top N%.

2. The arc originated from the syllable or word-syllable recognizer (and not from the word or fragment recognizer), and the OOV-region estimator score of the region belonged to the bottom N%.

The first rule is based on the assumption that the word recognizer's output will contribute only to IV query detection. The second rule is based on the assumption that the syllable and word-syllable recognizer's outputs will contribute only to OOV query detection. We assume that the fragment recognizer will have an intermediate property and will contribute to detecting both IV and OOV queries.

The unit selection method selects an optimum subword unit for an index according to the OOV-region estimator score. The unit selection works utterance-by-utterance. The assumption behind this method is that if we know about the absence of OOV in an utterance, we can use a coarser unit for an index. In our experiment, we selected the index unit according to the rules below.

1. If the maximum OOV-region estimator score obtained from an utterance was smaller than the threshold θ, the syllable-based transition network is used to represent the utterance.

2. Otherwise, the phoneme-based transition network is used.

[2]Many researchers call the OOV-region estimator the "OOV detector" in their papers, but we feel this term is confusable with the detection of OOV queries in STD. We therefore call this technique the "OOV-region estimator" in this study.

Note that most Japanese syllables consist of two phonemes, and therefore the syllable-based index tends to have significantly fewer arcs than the phoneme-based index. Because we need to compare detection scores from the phoneme-based and syllable-based indices, we normalized the scores by using the ratio between the syllable recognition rate and the phoneme recognition rate with development data.

6.3.2 OOV-Region Estimator

We implemented an OOV-region estimator similar to the one proposed by Parada et al. [88]. Bins of confusion networks provided from a word-syllable recognizer were treated as a classification unit. A conditional random field (CRF) trained with various features estimated the existence probability of OOVs in each bin of the confusion networks. The same as in [88], we used the subword existence probability $P_s(t_j)$ and the word entropy $H_w(t_j)$ as the features of the CRF.

$$P_s(t_j) = \sum_{s \in t_j} p(s|t_j), \tag{6.5}$$

$$H_w(t_j) = -\sum_{w \in t_j} p(w|t_j) \log p(w|t_j). \tag{6.6}$$

Here, t_j indicates the current bin of the confusion networks. Variables s and w indicate the syllables and words in each bin. Note that each bin of a confusion network could contain both words and syllables, and $\sum_{s \in t_j} p(s|t_j) + \sum_{w \in t_j} p(w|t_j) = 1$. In addition to these features, we used the word-syllable-mixed entropy $H_{ws}(t_j)$ as follows.

$$H_{ws}(t_j) = -\sum_{w \in t_j} p(w|t_j) \log p(w|t_j) - \sum_{s \in t_j} p(s|t_j) \log p(s|t_j). \tag{6.7}$$

Furthermore, we used the following features: best recognized word and its confidence, the difference of language-model scores between word and syllable recognizers, and the difference of acoustic-model scores between word and syllable recognizers.

6.4 Experimental Evaluation

6.4.1 Dataset

An evaluation was conducted using 39 hours of speech from the Corpus of Spontaneous Japanese (CSJ) [90], which contains 177 recordings of lectures. We used 46 hours of speech (200 lectures) from the CSJ as development data to make an OOV-region estimator. The

Table 6.1: Word and phoneme accuracy of speech recognizer.

Recognizer	Word Acc. (%)	Phoneme Acc. (%)
Word	74.5	88.6
Syllable	-	84.9
Word-Syllable	70.8	88.7
Fragment	-	87.9

rest of the CSJ (522 hours of speech) was used as training data for an acoustic model and language models. Evaluation, development, and training data had no overlap so as to ensure an open condition. A word dictionary was constructed from words that occurred more than three times in the training data. This resulted in a vocabulary size of 33,337 items. There were 2.00% and 2.04% of OOVs in the evaluation data and the development data, respectively.

We used a query set designed for the NTCIR-9 STD task [91] containing 50 IV queries (occurring 14.5 times on average) and 50 OOV queries (occurring 4.7 times on average). An F-measure (harmonic mean of precision and recall) averaged by queries was used as a measure of search accuracy. The detection threshold was varied and selected so as to maximize the F-measure.

Table 6.1 shows the word and phoneme accuracy of the recognizers described in Section 6.2. We used Julius [73] as the speech recognition engine. The syllable recognizer produced a slightly worse result. The other recognizers had almost the same phoneme accuracy.

6.4.2 Evaluation of OOV-Region Estimator

Figure 6.3 shows the false alarm rate (FA) and miss rate (Miss) of the OOV-region estimator we implemented. The FA indicates the ratio between the number of IVs detected as OOV and the actual number of IVs. The miss indicates the ratio between the number of not-detected OOVs and the actual number of OOVs. For example, we could detect about 70% of the OOV-region (30% Miss) with about 20% of false alarms.

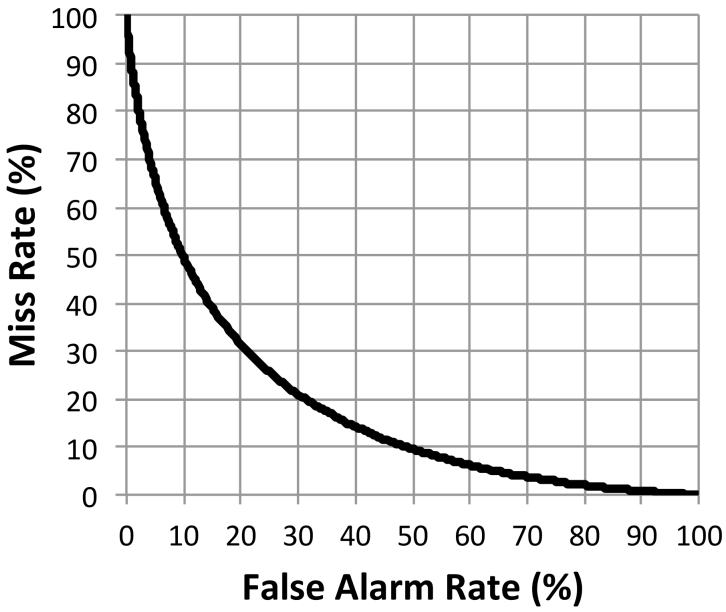


Figure 6.3: Evaluation of OOV-region estimator.

6.4.3 Evaluation of Index from Single Recognizer

Table 6.2 shows the results obtained by using the index from the single recognizer described in Section 6.2. The column "Recognizer" shows which type of recognizer was used, "Index unit" shows which type of subword unit was used in the index, "IV" and "OOV" show the F-measure for the IV and OOV queries, and "Index size" indicates the average number of arcs per word.

Results obtained from the 1-best index, confusion network index (CN) and transition network (TN) are shown for various representative conditions. The other rows indicate results from the 1-best index. A Word CN was created in the same manner described in [33], and a Phoneme TN was created from the 5-best hypotheses by using the method described in Section 6.2. As shown in Table 6.2, the confusion networks did not always improve accuracy. This is because the approximate matching we used produced too many false positives due to using the confusion network.

The best F-measure for the IV queries was obtained by using the word confusion network (CN) created from the word recognizer's output (78.2%). However, as expected, this method had a poor F-measure (35.4%) for the OOV queries[3]. The best F-measure

[3]Some compound words in OOV queries are designed to consist of OOV and IV words, and these were detected by using an approximate search. This is why the word-based method had an F-measure greater than 0.

Table 6.2: F-measure and index size of single system.

Recognizer	Index Unit	IV(%)	OOV(%)	Index Size
Word	Word (1-best/CN)	74.9/ **78.2**	19.7/ 35.4	1.03/ 5.90
	Syllable	76.4	46.8	1.81
	Phoneme (1-best/TN)	75.9/ 77.7	53.9/ 56.7	3.21/ 4.78
Syllable	Syllable	58.5	55.7	1.78
	Phoneme	61.5	62.8	3.16
Word-Syllable	Word	72.5	19.8	1.06
	Syllable	74.7	57.8	1.80
	Phoneme (1-best/TN)	74.9/ 76.9	**63.3**/ 62.0	3.20/ 4.68
Fragment	Syllable	68.0	55.3	1.80
	Phoneme (1-best/TN)	71.2/ 71.6	**63.3**/ 62.3	3.19/ 4.92

Table 6.3: F-measure and index size of combined system.

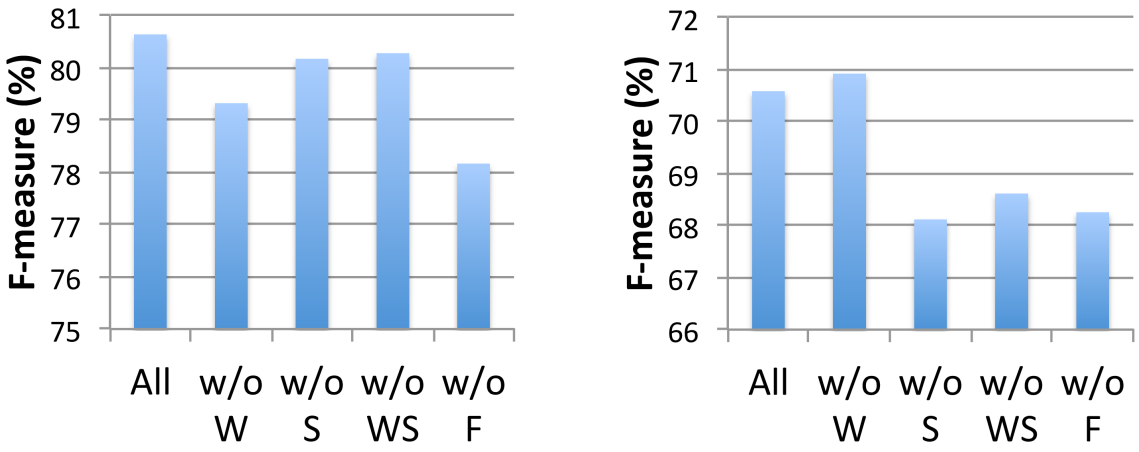
Recognizer	Index Unit	IV(%)	OOV(%)	Index Size
All	Syllable	**81.6**	65.0	2.35
	Phoneme	80.6	**70.6**	3.87

for the OOV queries was obtained by using the phoneme-based index from the fragment recognizer (63.3%). The phoneme-based index from the word recognizer produced much worse results (56.7%). Compared with the syllable-based index, the phoneme-based index always produced more accurate results: it had about 3.2 arcs per word, which was always more than the syllable-based index.

6.4.4 Evaluation of Index Combination

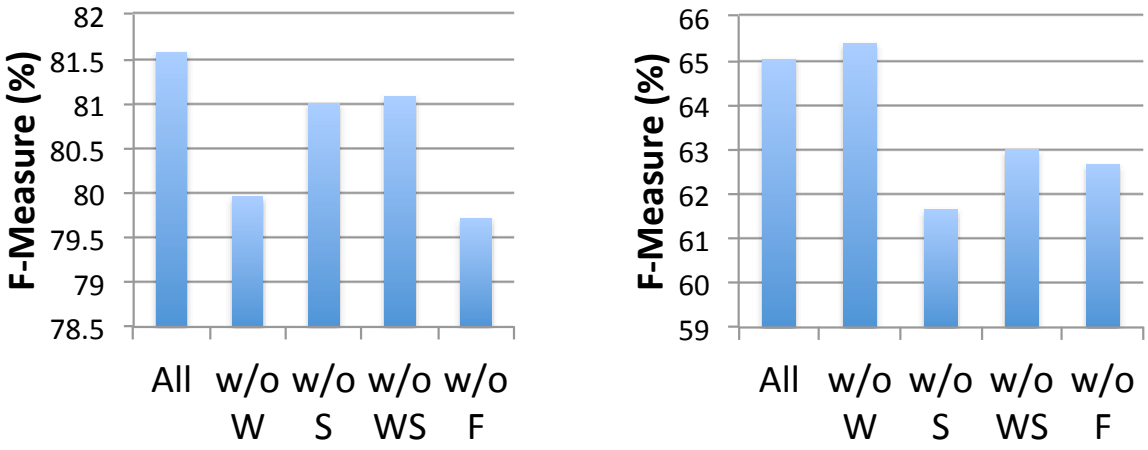
Table 6.3 shows the results obtained by using the combined transition network. The first column shows which recognizers ' outputs were combined. The remaining columns are the same as those in Table 6.2.

The syllable transition network showed improvement, especially for IV queries (81.6% of F-measure); however, accuracy for OOV queries was still low. The phoneme transition network achieved high F-measures for both IV and OOV queries (80.6% and 70.6%,



(a) IV query (b) OOV query

Figure 6.4: F-measure of phoneme transition networks without a specific recognizer (W: Word, S: Syllable, WS: Word-Syllable, F: Fragment).

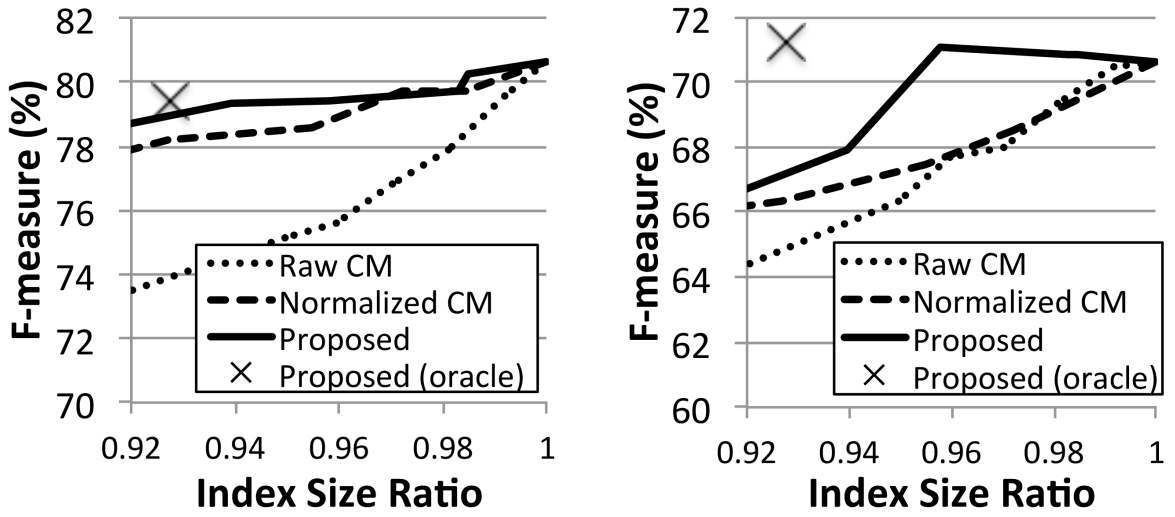


(a) IV query (b) OOV query

Figure 6.5: F-measure of syllable transition networks without a specific recognizer (W: Word, S: Syllable, WS: Word-Syllable, F: Fragment).

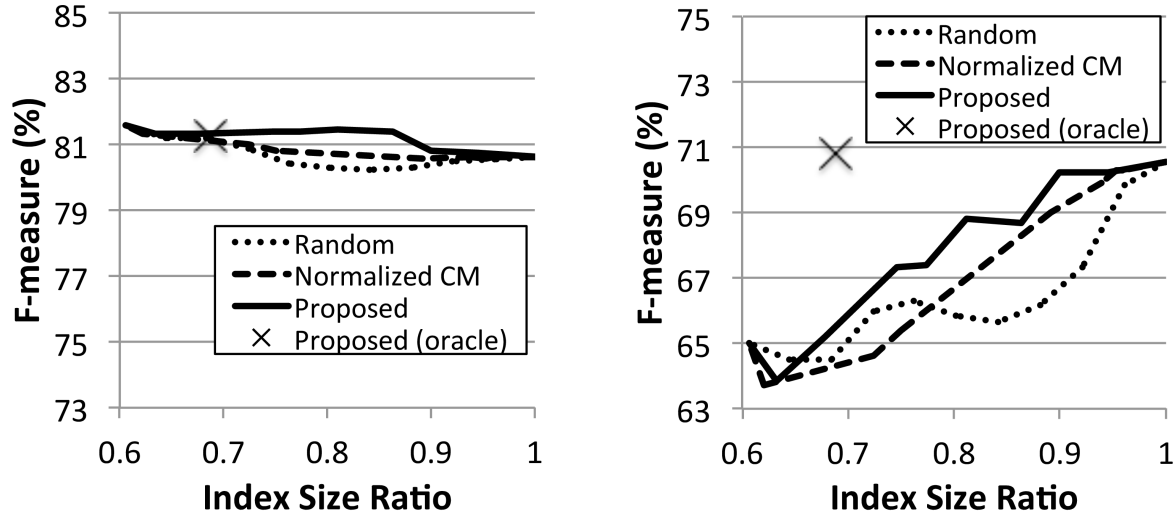
respectively); however, it had a relatively large index size (3.87 arcs per word).

On the basis of these observations, in the remainder of this study we regard the phoneme TN as the baseline method and focus on the index reduction of the phoneme TN. We next evaluated the contribution of each speech recognizer in terms of accuracy improvement by checking the accuracy when removing the single recognizer's results from the TN. The results are shown in Figure 6.4. Interestingly, the best F-measure for the OOV queries (70.9%) was obtained from the phoneme transition network without using a



(a) IV query (b) OOV query

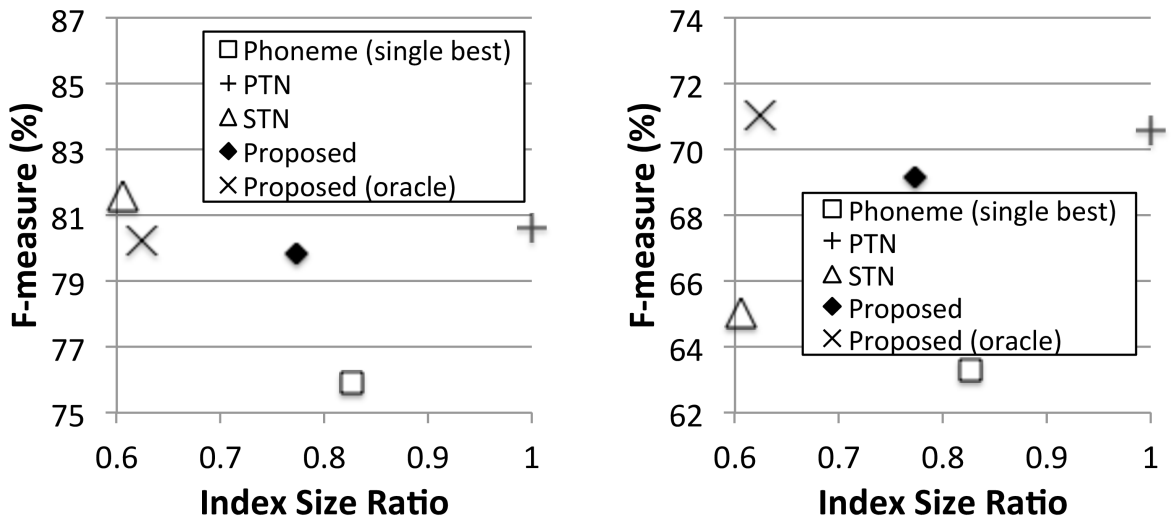
Figure 6.6: Evaluation of arc selection method.



(a) IV query (b) OOV query

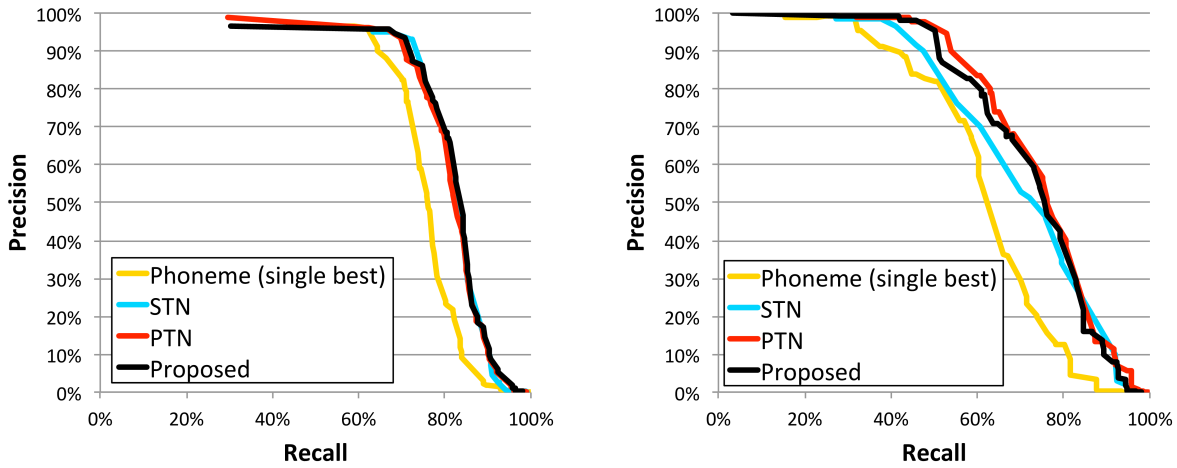
Figure 6.7: Evaluation of unit selection method.

word recognizer. This suggests that the index from the word recognizer only increased the false positives when detecting OOV queries. In a similar analysis, the syllable recognizer and word-syllable recognizer seemed to contribute little to detecting IV queries. The fragment recognizer was promising: it improved accuracy for both IV and OOV queries. The same trends were observed in experiments with syllable transition networks (Figure 6.5).



(a) IV query (b) OOV query

Figure 6.8: Evaluation of mixed method.



(a) IV query (b) OOV query

Figure 6.9: Recall-Precision curve.

6.4.5 Evaluation of Index Selection

Evaluation of Arc Selection Method

We first evaluated the arc selection method. The four methods below were compared.

- **Raw CM:** The method to remove arcs that had a confidence measure[4] below a certain threshold. If an arc corresponded to multiple speech recognizers, we used

[4] A confidence measure [49] from Julius [73] was used.

the highest confidence measure among those recognizers.

- **Normalized CM:** The method to remove arcs that had a normalized confidence measure below a certain threshold. The normalized confidence measure was calculated by using a logistic regression model with several features: a confidence measure from each recognizer[5], an inverse of the number of recognizers corresponding to the arc, and the number of arcs in the bin. The logistic regression model was trained using the development data.

- **Proposed:** The arc selection method based on the score of the OOV-region estimator.

- **Proposed (Oracle):** The arc selection method with correct OOV regions.

F-measures with various points of N are shown in Figure 6.6. We first observed that the confidence-measure-based method (Raw CM) severely degraded accuracy even for IV terms as arcs were removed. This indicates the difficulty of using confidence measures from different recognizers. The method using the normalized confidence measure (Normalized CM) seemed to be promising for IV query: the degradation of F-measure became much smaller than Raw CM. However, it still indicated severe degradation of accuracy for OOV query according to the index size reduction. It would because the region containing OOV queries normally produced low confidence, and the confidence-measure-based method removed the essential arcs in such regions. The proposed arc selection method worked better and could remove 4.2% of arcs without degrading accuracy. The oracle OOV-region estimator provided even better results, especially for OOV queries.

Evaluation of Unit Selection Method

Next, we evaluated the unit selection method.

- **Random:** The method to randomly select the phoneme TN or syllable TN for each utterance.

- **Normalized CM:** The method to select the syllable TN if the minimum of the normalized CM in the utterance was higher than a threshold. Otherwise, the phoneme TN was selected.

[5]If the arc was not produced by the recognizer, the value was set to 0.

- **Proposed:** The unit selection method based on the score of the OOV-region estimator.
- **Proposed (Oracle):** The unit selection method with correct OOV-regions.

Because we could not find any previous works that obviously relate to the unit selection method, we evaluated a random selection of the index unit (shown as "Random") and a selection method based on normalized confidence measure ("Normalized CM") as a reference.

F-measures with various points of threshold θ are shown in Figure 6.7. The OOV-region estimator-based method showed much better results than random selection or normalized confidence measure-based selection, achieving an 18.9% reduction of index size with only a slight degradation of accuracy. The oracle OOV-region estimator again produced a much better result for OOV queries.

Evaluation of Combined Method

Finally, we combined the arc selection method ($N = 60\%$; at the point of 4.2% index reduction in Figure 6.6) and the unit selection method ($\theta = 0.06\%$; at the point of 18.9% index reduction in Figure 6.7). Figure 6.8 shows the results from various systems. The proposed method had F-measures of 79.8% and 69.2% for IV and OOV queries, respectively. At this point, the proposed method achieved a 22.7% reduction of the index size from the combined phoneme transition network. Compared with the phoneme-based index from a single recognizer (maximum 75.9% for IV and 63.3% for OOV at the point of similar index size), the proposed method improved the F-measure by 3.9 and 5.9 points (16.3% and 16.0% relative error reduction, respectively) without increasing index size.

Figure 6.9 shows the precision-recall curve of various methods. The vertical and horizontal lines indicate the precision and recall, respectively. The results of PTN and of the proposed method are almost the same at each point. Therefore, we concluded that the proposed method could maintain the high accuracy of PTN while reducing the index size.

6.5 Summary

In this chapter, we proposed a novel index combination method for STD. Outputs from four different recognizers (word, syllable, word-syllable, and fragment recognizer) were combined. Two index-selection methods based on an OOV-region estimator were then

introduced and achieved a 22.7% reduction in index size while maintaining the high accuracy of the combined index. Compared with the best phoneme-based index from a single recognizer, the proposed method achieved relative error reductions of 16.3% and 16.0% for IV and OOV queries, respectively, without increasing the index size.

Chapter 7

Conclusion

In this thesis, we investigated many of the problems facing open-ended spoken language technologies based on two major applied systems: spoken dialogue systems and spoken document retrieval systems. First, we tackled the problem of spoken language analysis without specific-domain knowledge. A language analysis method of this type is necessary for covering a broad range of utterances, but we need to investigate a more precise and robust method by using only domain-independent information. Second, we tackled the problem of knowledge integration for the system to incorporate arbitrary knowledge. The system normally becomes more complex and more difficult to maintain as new knowledge is incorporated, and we therefore need to investigate the domain-extensible integration method.

In the remainder of this chapter, we summarize the contributions of this study and then discuss the remaining issues with suggestions for future directions.

7.1 Contributions of the Thesis

The contributions of this thesis are summarized as follows. The first two items relate to spoken dialogue systems and the latter two are for spoken document retrieval systems.

- We proposed new general context models for dialogues in database search tasks. The proposed context models differ from conventional ones in that they are domain-independent and can therefore be used even when the background database is changed. We proposed a spoken language understanding method based on these context models and demonstrated its highly accurate performance and robustness for speech recognition errors. More importantly, the proposed spoken language understanding method works even when replacing the background database.

- We proposed an extensible integration method of many spoken dialogue systems based on the new domain selection method. The proposed method differs from conventional ones in that it is designed to select whether the system should keep a current topic or not. This design enables the domain selection procedure to be domain-independent and allows for an extensible architecture in which a new domain expert can be created and added to the system in an extensible manner. Experimental results indicated that the proposed method not only achieved domain extensibility but also had more accurate results than conventional domain selection methods.

- This thesis proposes fast and accurate open vocabulary spoken term detection systems based on a new multi-stage rescoring algorithm. In this algorithm, we proposed combining several spoken term detection systems in tandem and narrowing down the search space in a stepwise fashion. Experimental results indicated that the proposed method was faster and significantly more accurate than conventional systems, especially when searching for unknown keywords.

- This thesis proposes an efficient integration method of many speech indices based on the new index selection methods. Many index combination methods with highly accurate search results have already been proposed, but with these methods the indices become too big as the combinations increase. This is problematic because an increased index size leads to both a slow search speed and a high storage cost. We therefore focused on developing an efficient index combination method that would not increase the index size. We advocate selectively combining only valuable indices and proposed new selection criteria based on an out-of-vocabulary region estimator. The proposed method achieved high search accuracy while suppressing the increase of index size.

As shown above, we successfully developed both spoken dialogue systems and spoken document retrieval systems that have the capabilities needed for open-ended spoken language systems. We hope that these works will function as a basis of open-ended spoken language technologies so that more and more spoken language systems can become commonly used.

7.2 Future Work

While this thesis has investigated many topics for open-ended spoken language technologies, there are still many important topics that remain unexplored. We conclude this thesis by listing these topics and suggesting future research directions. Note that while the topics below are mainly derived from the two systems discussed in this thesis, they are applicable to numerous other systems, as well.

7.2.1 Topics on Spoken Dialogue Systems

Information sharing model among domain experts In this thesis, we proposed a distributed architecture in which many domain experts work in parallel. Although we implemented a simple message passing protocol to share geographical information, we believe there are more consistent models to share dialogue context among domain experts. The information needs to be consistent among domain experts, including newly created ones. Studies in this direction could be related to ontology and formal semantics. Realistically speaking, spoken dialogue systems working on single domain knowledge are still being studied very actively for realizing more sophisticated spoken language understanding methods and dialogue management. Any new context model should be expressive enough to be used with these techniques.

Optimal design on granularity of domain knowledge While we have developed a tourist information system by combining five domain experts, we have not discussed the optimality of the granularity of each domain. For example, using finer-grained domains would make each spoken language analysis method simpler, and at the same time it would make the integration method more complex. With coarser-grained domains, the opposite would be true. Therefore, there would be optimal granularity of domains in terms of a system development. Discussion these details should prove useful for designing a broad range of open-ended systems.

Extension for multi-modal systems In this thesis, we primarily focused on handling spoken language input, but there are many other modalities that could function as the input of a natural man-machine interface, such as gestures or gaze information. One representative research area is robotics, in which many types of sensors need to be handled.

Agent systems on mobile phones or car navigation systems are also actively studied in this direction. While we have already started research on combining multiple experts reliable for different modalities in a robot [13, 15], there is still plenty of room for investigation.

7.2.2 Topics on Spoken Document Retrieval Systems

Extension for spoken content retrieval systems In this thesis, we mainly focused on keyword detection from speech databases and did not touch on the detection of recorded segments semantically related to the keywords. This task is actively studied as spoken content retrieval [26, 27]. It is more complex than spoken term detection because it requires not only detecting out-of-vocabulary words but also associating these words with each other. Studies in this direction, especially when aimed at open-ended systems, need a consistent expression of semantics. Fortunately, there have already been many studies and repositories for semantics, including lexical semantic repositories (e.g., WordNet [92], VerbNet [93], FrameNet [94], etc.) and ontological repositories (e.g., YAGO2 [95], ConceptNet [96], FreeBase [97], etc.). Therefore, studies that link these works with spoken documents would be a promising area for future research.

Reducing the computational cost of indexing In this thesis, we focused on reducing the size of combined indices to quicken the search speed and reduce the storage cost. However, the computational cost of indexing is also an important factor of spoken document retrieval systems because it directly affects the maintenance cost of the indexing servers. Normally, the computational cost of indexing increases in accordance with the number of combined indices, so techniques to reduce the computational cost should be pursued. Studies in this direction could focus on lightweight domain selection before indexing or creating an optimization framework of whole systems.

Bibliography

[1] Patti Price. Evaluation of spoken language systems: The atis domain. In *Proceedings of the Third DARPA Speech and Natural Language Workshop*, pages 91–95. Morgan Kaufmann, 1990.

[2] Kazunori Komatani, Shinichi Ueno, Tatsuya Kawahara, and Hiroshi G. Okuno. Flexible guidance generation using user model in spoken dialogue systems. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 256–263. Association for Computational Linguistics, 2003.

[3] Antoine Raux, Brian Langner, Dan Bohus, Alan W. Black, and Maxine Eskenazi. Let 's go public! taking a spoken dialog system to the real world. In *in Proc. of Interspeech 2005*. Citeseer, 2005.

[4] Dave Abberley, Steve Renals, and Gary Cook. Retrieval of broadcast news documents with the thisl system. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 6, pages 3781–3784. IEEE, 1998.

[5] John S. Garofolo, Cedric G. P. Auzanne, and Ellen M. Voorhees. The trec spoken document retrieval track: A success story. In *in Text Retrieval Conference (TREC) 8*, pages 16–19, 2000.

[6] Bor-shen Lin, Hsin-min Wang, and Lin-shan Lee. A distributed agent architecture for intelligent multi-domain spoken dialogue systems. In *IEICE Trans. on Information and Systems*, E84-D(9), pages 1217–1230, Sept. 2001.

[7] Norihito Yasuda, Kohji Dohsaka, Kiyoaki Aikawa, and Shinichi Ueno. Developing a multi-domain dialogue system by the integration of single-domain systems (in Japanese). In *IPSJ SIG Technical Report*, 2003-SLP-45-20, 2003.

[8] Makoto Nagamori, Nobuo Kawaguchi, Shigeki Matsubara, Katsuhiko Toyama, and Yasuyoshi Inagaki. A framework for multi - domain conversational systems (in Japanese). In *IPSJ SIG Technical Report*, 2000-SLP-31-7, 2000.

[9] Nobuo Kawaguchi, Makoto Nagamori, Shigeki Matsubara, and Yasuyoshi Inagaki. Multi — domain spoken dialog system speech recognition unified management architecture (in Japanese). In *IPSJ SIG Technical Report*, 2001-SLP-36-10, 2001.

[10] Ian O'Neill, Philip Hanna, Xingkun Liu, and Michael McTear. Cross domain dialogue modelling: An object-based approach. In *Proc. ICSLP*, volume I, 2004.

[11] Botond Pakucs. Towards dynamic multi-domain dialogue processing. In *Proc. Eurospeech*, pages 741–744, 2003.

[12] Noboru Miyazaki, Tetsuo Amakasu, Akihiro Tomihisa, and Teruo Hagino. A semi-automatic dialogue system integration method for handling multi-domain dialogue (in Japanese). In *the 2005 Autumn Meeting of ASJ*, pages 189–190, 2005.

[13] Mikio Nakano, Yuji Hasegawa, Toyotaka Torii, Yohane Takeuchi, Kazuhiro Nakadai, Hiroshi Tsujino, Naoyuki Kanda, and Hiroshi G. Okuno. A two-layer model for behavior and dialogue planning in conversational service robots. In *Proc. IROS*, pages 1542–1548, Aug. 2005.

[14] Stephanie Seneff, Edward Hurley, Raymond Lau, Christine Pao, Philipp Schmid, and Victor Zue. Galaxy-ii: a reference architecture for conversational system development. In *ICSLP*, volume 98, pages 931–934, 1998.

[15] Mikio Nakano, Yuji Hasegawa, Kotaro Funakoshi, Johane Takeuchi, Toyotaka Torii, Kazuhiro Nakadai, Naoyuki Kanda, Kazunori Komatani, Hiroshi G Okuno, and Hiroshi Tsujino. A multi-expert model for dialogue and behavior control of conversational robots and agents. *Knowledge-Based Systems*, 24(2):248–256, 2011.

[16] Terry Winograd. Understanding natural language. *Cognitive Psychology*, 3(1):1–191, 1972.

[17] Victor Zue, James Glass, David Goodine, Hong Leung, Michael Phillips, Joseph Polifroni, and Stephanie Seneff. Integration of speech recognition and natural language

processing in the mit voyager system. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pages 713–716. IEEE, 1991.

[18] Christian Raymond and Giuseppe Riccardi. Generative and discriminative algorithms for spoken language understanding. In *INTERSPEECH*, pages 1605–1608, 2007.

[19] Steve Young. Using pomdps for dialog management. In *Spoken Language Technology Workshop, 2006. IEEE*, pages 8–13. IEEE, 2006.

[20] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[21] Ryuichi Nisimura, Akinobu Lee, Hiroshi Saruwatari, and Kiyohiro Shikano. Public speech-oriented guidance system with adult and child discrimination capability. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 1, pages I–433. IEEE, 2004.

[22] Jerome R. Bellegarda. Large—scale personal assistant technology deployment: the siri experience. In *Proc. INTERSPEECH*, pages 2029–2033, 2013.

[23] Kosuke Tsujino, Shinya Iizuka, Yusuke Nakashima, and Yoshinori Isoda. Speech recognition and spoken language understanding for mobile personal assistants: A case study of" shabette concier". In *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, volume 2, pages 225–228. IEEE, 2013.

[24] T. Isobe, S. Hayakawa, H. Murao, K. Mizutani, and F. Itakura. A study on domain recognition of spoken dialogue systems. In *Proc. EUROSPEECH*, 2003.

[25] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion. Results of the 2006 spoken term detection evaluation. In *Proceedings of ACM SIGIR Workshop on Searching Spontaneous Conversational Speech*, pages 51–55, 2007.

[26] T. Akiba, H. Nishizaki, K. Aikawa, T. Kawahara, and T. Matsui. Overview of the IR for spoken documents task in NTCIR-9 workshop. In *Proc. NTCIR-9*, 2011.

[27] Tomoyosi Akiba, Hiromitsu Nishizaki, Kiyoaki Aikawa, Xinhui Hu, Yoshiaki Itoh, Tatsuya Kawahara, Seiichi Nakagawa, Hiroaki Nanjo, and Yoichi Yamashita.

Overview of the ntcir-10 spokendoc-2 task. In *Proceedings of the 10th NTCIR Conference*, 2013.

[28] Steve Renals, Dave Abberleya, David Kirbyb, and Tony Robinson. Indexing and retrieval of broadcast news. *Speech Communication*, 32(1/2):5–20, 2000.

[29] Takaaki Hori, I. Lee Hetherington, Timothy J. Hazen, and James R. Glass. Open-vocabulary spoken utterance retrieval using confusion networks. In *Proc. ICASSP*, volume 4, pages 73–76, 2007.

[30] C. Allauzen, M. Mohri, and M. Saraclar. General Indexation of Weighted Automata–Application to Spoken Utterance Retrieval. *Proc. HLT*, pages 33–40, 2004.

[31] M. Saraclar and R. Sproat. Lattice-based search for spoken utterance retrieval. In *Proc. HLT-NAACL*, pages 129–136, 2004.

[32] P. Yu and F. Seide. A hybrid word/phoneme-based approach for improved vocabulary-independent search in spontaneous speech. *Proc. ICLSP '04*, 2004.

[33] T. Hori, I. L. Hetherington, T. J. Hazen, and J. R. Glass. Open-vocabulary spoken utterance retrieval using confusion networks. In *Proc. ICASSP*, volume 4, pages IV–73. IEEE, 2007.

[34] N. Kanda, H. Sagawa, T. Sumiyoshi, and Y. Obuchi. Open-vocabulary keyword detection from super-large scale speech database. In *Proc. MMSP*, pages 939–944. IEEE, 2008.

[35] C. Parada, A. Sethy, and B. Ramabhadran. Balancing false alarms and hits in spoken term detection. In *Proc. ICASSP*, pages 5286–5289. IEEE, 2010.

[36] Roy Wallace, Robbie Vogt, and Sridha Sridharan. A Phonetic Search Approach to the 2006 NIST Spoken Term Detection Evaluation. In *Proc. Interspeech*, pages 2385–2388, 2007.

[37] Yoshiaki Itoh, Takayuki Otake, Kohei Iwata, Kazunori Kojima, Masaaki Ishigame, Kazuyo Tanaka, and Shi wook Lee. Two-Stage Vocabulary-Free Spoken Document Retrieval — Subword Identification and Re-Recognition of the Identified Sections. In *Proc. Interspeech*, pages 1161–1164, 2006.

[38] Kohei Iwata, Yoshiaki Itoh, Kazunori Kojima, Masaaki Ishigame, Kazuyo Tanaka, and Shi wook Lee. Open-Vocabulary Spoken Document Retrieval Based on New Subword Models and Subword Phonetic Similarity. In *Proc. Interspeech*, pages 325–328, 2006.

[39] Kouichi Katsurada, Shigeki Teshima, and Tsuneo Nitta. Fast keyword detection using suffix array. In *INTERSPEECH*, pages 2147–2150, 2009.

[40] Naoyuki Kanda, Hirohiko Sagawa, Takashi Sumiyoshi, and Yasunari Obuchi. Open-vocabulary keyword detection from super-large scale speech database. In *MMSP*, pages 939–944, 2008.

[41] Peng Yu, Kaijiang Chen, Chengyuan Ma, and Frank Seide. Vocabulary-independent indexing of spontaneous speech. *Speech and Audio Processing, IEEE Trans.*, 13(5 Part 1):635–643, 2005.

[42] Seiichi Nakagawa, Keisuke Iwami, Yasuhisa Fujii, and Kazumasa Yamamoto. A robust/fast spoken term detection method based on a syllable n-gram index with a distance metric. *Speech Communication*, 2012.

[43] Satya Dharanipragada and Salim Roukos. A multistage algorithm for spotting new words in speech. *Speech and Audio Processing, IEEE Trans.*, 10(8):542–550, 2002.

[44] P. Yu and Frank Seide. A hybrid word/phoneme-based approach for improved vocabulary-independent search in spontaneous speech. *Eighth International Conference on Spoken Language Processing*, pages 293–296, 2004.

[45] S. Meng, P. Yu, J. Liu, and F. Seide. Fusing multiple systems into a compact lattice index for chinese spoken term detection. In *Proc. ICASSP*, pages 4345–4348. IEEE, 2008.

[46] I. Bufyko, O. Kimball, M.H. Siu, J. Herrero, and D. Blum. Detection of unseen words in conversational mandarin. In *Proc ICASSP*, pages 5181–5184. IEEE, 2012.

[47] H. Nishizaki, H. Furuya, S. Natori, and Y. Sekiguchi. Spoken term detection using multiple speech recognizers 'outputs at NTCIR-9 SpokenDoc STD subtask. In *Proc. NTCIR-9*, 2011.

[48] Y. Itoh, K. Iwata, M. Ishigame, K. Tanaka, and S. Lee. Spoken term detection results using plural subword models by estimating detection performance for each query. In *Proc. INTERSPEECH*, 2011.

[49] Akinobu Lee, Kiyohiro Shikano, and Tatsuya Kawahara. Real-time word confidence scoring using local posterior probabilities on tree trellis search. In *Proc. ICASSP*, volume 1, pages I–793. IEEE, 2004.

[50] Peng Yu, Yu Shi, and Frank Seide. Approximateword-lattice indexing with text indexers: Time-anchored lattice expansion. In *Proc. ICASSP*, pages 5248–5251. IEEE, 2008.

[51] Jie Gao, Qingwei Zhao, Yonghong Yan, and Jian Shao. Efficient system combination for syllable-confusion-network-based chinese spoken term detection. In *Proc. ISCSLP*, pages 1–4. IEEE, 2008.

[52] Joel Pinto, Igor Szoke, SRM Prasanna, and Hynek Hermansky. Fast approximate spoken term detection from sequence of phonemes. In *Proceedings of the ACM SIGIR Workshop on Searching Spontaneous Conversational Speech*, pages 08–45, 2008.

[53] Kazuyo Tanaka, Yoshiaki Itoh, Hiroaki Kojima, and Nahoko Fujimura. Speech data retrieval system constructed on a universal phonetic code domain. In *Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on*, pages 323–326. IEEE, 2001.

[54] Frank Seide, Peng Yu, Chengyuan Ma, and Eric Chang. Vocabulary-independent search in spontaneous speech. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 1, pages I–253. IEEE, 2004.

[55] Dan Moldovan, Marius Paşca, Sanda Harabagiu, and Mihai Surdeanu. Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems (TOIS)*, 21(2):133–154, 2003.

[56] Tomek Strzalkowski and Sanda M Harabagiu. *Advances in open domain question answering*, volume 32. Springer, 2006.

[57] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010.

[58] Kazunori Komatani and Tatsuya Kawahara. Flexible dialogue management for generating efficient confirmation and guidance using confidence measures of speech recognition result (in Japanese). *IPSJ Journal*, 43(10):3078–3986, 2002.

[59] Seiichi Nakagawa and Yoshihisa Horibe. Confidence measures for speech recognition by using likelihood of acoustic model and language model (in Japanese). In *IPSJ SIG Technical Report*, 2001-SLP-36-13, 2001.

[60] Akinobu Lee, Masashi Yamada, Ryuichi Nishimura, and Kiyohiro Shikano. Public speech-oriented information guidance system.

[61] Ryuichiro Higashinaka, Mikio Nakano, and Kiyoaki Aikawa. A statistical discourse understanding method for multiple-context-based spoken dialogue systems (in Japanese). In *IPSJ SIG Technical Report*, 2003-SLP-45-17, 2003.

[62] Yuki Irie, Shigeki Matsubara, Nobuo Kawaguchi, Yukiko Yamaguchi, and Yasuyoshi Inagaki. Speech intention understanding based on spoken dialogue corpus (in Japanese). In *JSAI Technical Report*, SIG-SLUD-A301-03, 2003.

[63] Malte Gabsdil and Oliver Lemon. Combining acoustic and pragmatic features to predict recognition performance in spoken dialogue systems. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 343–350, July 2004.

[64] Marilyn Walker, Irene Langkilde, Jerry Wright, Allen Gorin, and Diane Litman. Learning to predict problematic situations in a spoken dialogue system: Experiments with how may i help you? In *Proc. of 1st Conf. on the North American Chapter of ACL (NAACL00)*, pages 210–217, 2000.

[65] Hiroki Yuasa, Satoshi Mizuno, Toshihiko Itoh, Atsuhiko Kai, Tatsuhiro Konishi, and Yukihiro Itoh. Construction and evaluation of spoken dialogue type car interface using a situation and the context (in Japanese). In *IPSJ SIG Technical Report*, 2003-SLP-49-34, 2003.

[66] Sameer S. Pradhan and Wayne H. Ward. Estimating semantic confidence for spoken dialogue systems. In *Proc. ICASSP*, volume 1, pages 233–236, 2002.

[67] M. Araki, K. Komatani, T. Hirata, and S. Doshita. A dialogue library for task-oriented spoken dialogue systems. In *IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 1–7, 1999.

[68] V. Zue. Jupiter: A telephone-based conversational interface for weather information. In *IEEE Trans. on Speech and Audio Processing*, volume 8, January 2000.

[69] Naoyuki Kanda, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Spoken language understanding using dialogue context in database search task (in Japanese). *IPSJ Journal*, 47(6):1802–1811, 2006.

[70] Kazunori Komatani, Tatsuya Kawahara, Yoji Kiyota, Sadao Kurohashi, and Pascale Fung. Restaurant search system with speech interface using flexible language model and matching (in Japanese). In *IPSJ SIG Technical Report*, SLP-39-30, 2001.

[71] Jennifer Chu-Carroll and Bob Carpenter. Vector-based natural language call routing. *Computational Linguistics*, 25(3):361–388, 1999.

[72] *C5.0.* http://rulequest.com/index.html.

[73] A. Lee, T. Kawahara, and K. Shikano. Julius—an open source real-time large vocabulary recognition engine. In *Proc. EUROSPEECH*, pages 1691–1694, 2001.

[74] Akinobu Lee Hideki Banno Kazuya Takeda Masato Mimura Takeshi Yamada Takanobu Nishiura Katsunobu Ito Akinori Ito Kiyohiro Shikano Tatsuya Kawahara, Takashi Sumiyoshi. Product software of continuous speech recognition consortium : 2001 version (in Japanese). In *IPSJ SIG Technical Report*, 2002-SLP-43-3, 2002.

[75] Naoyuki Kanda, Kazunori Komatani, Mikio Nakano, Kazuhiro Nakadai, Hiroshi Tsujino, Tetsuya Ogata, and Hiroshi G. Okuno. Robust domain selection using dialogue history in multi-domain spoken dialogue systems (in Japanese). *IPSJ Journal*, 48(5):1980–1989, 2007.

[76] I. Lane, T. Kawahara, T. Matsui, and S. Nakamura. Topic classification and verification modeling for out-of-domain utterance detection. In *Proc. ICSLP*, pages 2197–2200, 2004.

[77] Ryuichiro Higashinaka, Katsuhito Sudoh, and Mikio Nakano. Incorporating Discourse Features into Confidence Scoring of Intention Recognition Results in Spoken Dialogue Systems. In *Proc. ICASSP*, volume 1, pages 25–28, 2005.

[78] Satoshi Ikeda, Kazunori Komatani, Tetsuya Ogata, Hiroshi G Okuno, and Hiroshi G Okuno. Extensibility verification of robust domain selection against out-of-grammar utterances in multi-domain spoken dialogue system. In *INTERSPEECH*, pages 487–490, 2008.

[79] Naoyuki Kanda, Kazunori Komatani, Mikio Nakano, Kazuhiro Nakadai, Hiroshi Tsujino, Tetsuya Ogata, and Hiroshi G. Okuno. Robust domain selection using dialogue history in multi-domain spoken dialogue system (in Japanese). In *IPSJ SIG Technical Report*, 2006-SLP60-11, 2006.

[80] Naoyuki Kanda, Takashi Sumiyoshi, Hiroaki Kokubo, Hirohiko Sagawa, and Yasunari Obuchi. Spoken term detection from large scale speech database using multistage rescoring (in Japanese). *IEICE Transaction on Information and Systems*, J95-D(4):969–981, 2012.

[81] Tatsuya Kawahara, Toshihiko Munetsugu, and Shuji Doshita. Word spotting in spontaneous speech with heuristic language model (in Japanese). *IECE Transaction on Information and Systems*, J78-D-2(7):1013–1020, 1995.

[82] Hiromitsu Nishizaki, Xinhui Hu, Hiroaki Nanjo, Yoshiaki Itoh, Tomoyosi Akiba, Tatsuya Kawahara, Seiichi Nakagawa, Tomoko Matsui, Yoichi Yamashita, and Kiyoaki Aikawa. Development of test collection for spoken term detection and its baseline evaluation (in Japanese). In *IPSJ SIG Technical Report*, volume 2010-SLP-81, page 8, 2010.

[83] K. Maekawa. Corpus of Spontaneous Japanese: Its design and evaluation. *Proc. ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, pages 7–12, 2003.

[84] Naoyuki Kanda, Katsutoshi Itoyama, and Hiroshi G. Okuno. Selective index combination method based on out-of-vocabulary region estimator for open-vocabulary spoken term detection (in Japanese). *IPSJ Journal*, 55(3), 2014 (accepted).

[85] O. Siohan and M. Bacchiani. Fast vocabulary-independent audio search using path-based graph indexing. In *Proc. INTERSPEECH*, pages 53–56, 2005.

[86] L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Arxiv preprint cs.CL/0010012*, 2000.

[87] A. Rastrow, A. Sethy, and B. Ramabhadran. A new method for OOV detection using hybrid word/fragment system. In *Proc. ICASSP*, pages 3953–3956. IEEE, 2009.

[88] C. Parada, M. Dredze, D. Filimonov, and F. Jelinek. Contextual information improves OOV detection in speech. In *Proc. NAACL-HLT*, pages 216–224. Association for Computational Linguistics, 2010.

[89] L. Qin, M. Sun, and A. Rudnicky. System combination for out-of-vocabulary word detection. In *Proc. ICASSP*, pages 4817–4820. IEEE, 2012.

[90] K. Maekawa. Corpus of spontaneous japanese: Its design and evaluation. In *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[91] Y. Itoh, H. Nishizaki, X. Hu, H. Nanjo, T. Akiba, T. Kawahara, S. Nakagawa, T. Matsui, Y. Yamashita, and K. Aikawa. Constructing japanese test collections for spoken term detection. In *Proc INTERSPEECH*, 2010.

[92] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[93] Karin Kipper Schuler. Verbnet: A broad-coverage, comprehensive verb lexicon. 2005.

[94] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998.

[95] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[96] Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004.

[97] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

Relevant Publications

Chapters 3

1. Naoyuki Kanda, Kazunori Komatani, Tetsuya Ogata and Hiroshi G. Okuno, "Spoken Language Understanding Using Dialogue Context in Database Search Task," *IPSJ Journal*, Vol. 47, No. 6, pp. 1802–1811, June 2006 (in Japanese).

2. Kazunori Komatani, Naoyuki Kanda, Tetsuya Ogata and Hiroshi G. Okuno, "Contextual constraints based on dialogue models in database search task for spoken dialogue systems," in *Proc. of the 9th European Conference on Speech Communication and Technology (INTERSPEECH 2005)*, pp. 877–880, September 2005.

Chapter 4

1. Naoyuki Kanda, Kazunori Komatani, Mikio Nakano, Kazuhiro Nakadai, Hiroshi Tsujino, Tetsuya Ogata and Hiroshi G. Okuno, "Robust Domain Selection Using Dialogue History in Multi-domain Spoken Dialogue Systems," *IPSJ Journal*, Vol. 48, No. 5, pp. 1980–1989, May 2007 (in Japanese).

2. Mikio Nakano, Yuji Hasegawa, Kotaro Funakoshi, Johane Takeuchi, Toyotaka Torii, Kazuhiro Nakadai, Naoyuki Kanda, Kazunori Komatani, Hiroshi G Okuno, Hiroshi Tsujino, "A multi-expert model for dialogue and behavior control of conversational robots and agents," *Knowledge-Based Systems*, Vol. 24, No. 2, pp. 248–256, March 2011.

3. Kazunori Komatani, Naoyuki Kanda, Mikio Nakano, Kazuhiro Nakadai, Hiroshi Tsujino, Tetsuya Ogata and Hiroshi G. Okuno, "Multi-domain spoken dialogue system with extensibility and robustness against speech recognition errors," in *Proc. of the 7th SIGdial Workshop on Discourse and Dialogue (SIGdial 2009)*, pp. 9–17, July 2009.

4. Mikio Nakano, Yuji Hasegawa, Kazuhiro Nakadai, Takahiro Nakamura, Johane Takeuchi, Toyotaka Torii, Hiroshi Tsujino, Naoyuki Kanda and Hiroshi G. Okuno, "A two-layer model for behavior and dialogue planning in conversational service robots," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pp. 3329–3335, August 2005.

Chapter 5

1. Naoyuki Kanda, Takashi Sumiyoshi, Hiroaki Kokubo, Hirohiko Sagawa and Yasunari Obuchi, "Spoken Term Detection from Large Scale Speech Database Using Multistage Rescoring," *IEICE Transaction on Information and Systems*, Vol. J95-D, No. 4, pp. 969–981, April 2012 (in Japanese).

2. Naoyuki Kanda, Ryu Takeda and Yasunari Obuchi, "Using rhythmic features for Japanese spoken term detection," in *Proc. of the 4th IEEE Workshop on Spoken Language Technology (SLT 2012)*, pp. 170–175, December 2012.

3. Naoyuki Kanda, Takashi Sumiyoshi, Hiroaki Kokubo, Hirohiko Sagawa and Yasunari Obuchi, "Open-vocabulary keyword detection from super-large scale speech database," in *Proc. of the 10th International IEEE International Workshop on Multimedia Signal Processing (MMSP 2008)*, pp. 939–944, October 2008.

Chapter 6

1. Naoyuki Kanda, Katsutoshi Itoyama and Hiroshi G. Okuno, "Selective Index Combination Method based on Out-of-Vocabulary Region Estimator for Open-Vocabulary Spoken Term Detection," *IPSJ Journal*, Vol. 55, No. 3, March 2014 (in Japanese) (accepted).

2. Naoyuki Kanda, Katsutoshi Itoyama and Hiroshi G. Okuno, "Multiple index combination for Japanese spoken term detection with optimum index selection based on OOV-region classifier," in *Proc. of the 32nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, pp. 8540–8544, May 2013.

All Publications by the Author

Major Publications

Journal Papers

1) Naoyuki Kanda, Kazunori Komatani, Tetsuya Ogata and Hiroshi G. Okuno, "Spoken Language Understanding Using Dialogue Context in Database Search Task," *IPSJ Journal*, Vol. 47, No. 6, pp. 1802–1811, June 2006 (in Japanese).

2) Naoyuki Kanda, Kazunori Komatani, Mikio Nakano, Kazuhiro Nakadai, Hiroshi Tsujino, Tetsuya Ogata and Hiroshi G. Okuno, "Robust Domain Selection Using Dialogue History in Multi-domain Spoken Dialogue Systems," *IPSJ Journal*, Vol. 48, No. 5, pp. 1980–1989, May 2007 (in Japanese).

3) Naoyuki Kanda, Takashi Sumiyoshi, Hiroaki Kokubo, Hirohiko Sagawa and Yasunari Obuchi, "Spoken Term Detection from Large Scale Speech Database Using Multistage Rescoring," *IEICE Transaction on Information and Systems*, Vol. J95-D, No. 4, pp. 969–981, April 2012 (in Japanese).

4) Naoyuki Kanda, Katsutoshi Itoyama and Hiroshi G. Okuno, "Selective Index Combination Method based on Out-of-Vocabulary Region Estimator for Open-Vocabulary Spoken Term Detection," *IPSJ Journal*, Vol. 55, No. 3, March 2014 (in Japanese) (accepted).

5) Mikio Nakano, Yuji Hasegawa, Kotaro Funakoshi, Johane Takeuchi, Toyotaka Torii, Kazuhiro Nakadai, Naoyuki Kanda, Kazunori Komatani, Hiroshi G Okuno, Hiroshi Tsujino, "A multi-expert model for dialogue and behavior control of conversational robots and agents," *Knowledge-Based Systems*, Vol. 24, No. 2, pp. 248–256, March 2011.

Letters

6) Naoyuki Kanda, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno, "Management of Query Conditions using Dialogue History in Spoken Dialogue Systems," *Information Technology Letters*, Vol. 3, pp. 131-132, September 2004 (in Japanese).

Book Chapters

7) Hisashi Ikeda, Nobuo Nukaga, Yoshiyuki Kobayashi, Naoyuki Kanda, Yuki Watanabe, and Junichi Hirayama, "Media Processing Technologies for Repurposing Large-Scale Unstructured Data", *Journal of the Japansese Society for Artificial Intelligence*, Vol. 28, No. 1, pp. 114–121, January 2013 (in Japanese).

International Conference Papers (Refereed)

8) Naoyuki Kanda, Takashi Sumiyoshi, Hiroaki Kokubo, Hirohiko Sagawa and Yasunari Obuchi, "Open-vocabulary keyword detection from super-large scale speech database," in *Proc. of the 10th International IEEE International Workshop on Multimedia Signal Processing (MMSP 2008)*, pp. 939–944, October 2008.

9) Naoyuki Kanda, Ryu Takeda and Yasunari Obuchi, "Using rhythmic features for Japanese spoken term detection," in *Proc. of the 4th IEEE Workshop on Spoken Language Technology (SLT 2012)*, pp. 170–175, December 2012.

10) Naoyuki Kanda, Katsutoshi Itoyama and Hiroshi G. Okuno, "Multiple index combination for Japanese spoken term detection with optimum index selection based on OOV-region classifier," in *Proc. of the 32nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, pp. 8540–8544, May 2013.

11) Naoyuki Kanda, Ryu Takeda and Yasunari Obuchi, "Noise Robust Speaker Verification with Delta Cepstrum Normalization," in *Proc. of the 14th European Conference on Speech Communication and Technology (INTERSPEECH 2013)*, pp. 3112–3116, August 2013.

12) Naoyuki Kanda, Ryu Takeda and Yasunari Obuchi, "Elastic Spectral Distortion for Low Resource Speech Recognition with Deep Neural Networks," in *Proc. of the*

13th biannual IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2013), pp. 309–314, December 2013.

13) Mikio Nakano, Yuji Hasegawa, Kazuhiro Nakadai, Takahiro Nakamura, Johane Takeuchi, Toyotaka Torii, Hiroshi Tsujino, Naoyuki Kanda and Hiroshi G. Okuno, "A two-layer model for behavior and dialogue planning in conversational service robots," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pp. 3329–3335, August 2005.

14) Kazunori Komatani, Naoyuki Kanda, Tetsuya Ogata and Hiroshi G. Okuno, "Contextual constraints based on dialogue models in database search task for spoken dialogue systems," in *Proc. of the 9th European Conference on Speech Communication and Technology (INTERSPEECH 2005)*, pp. 877–880, September 2005.

15) Kazunori Komatani, Naoyuki Kanda, Mikio Nakano, Kazuhiro Nakadai, Hiroshi Tsujino, Tetsuya Ogata and Hiroshi G. Okuno, "Multi-domain spoken dialogue system with extensibility and robustness against speech recognition errors," in *Proc. of the 7th SIGdial Workshop on Discourse and Dialogue (SIGdial 2009)*, pp. 9–17, July 2009.

16) Yasunari Obuchi, Ryu Takeda and Naoyuki Kanda, "Voice activity detection based on augmented statistical noise suppression," in *Proc. of the 4th Annual Conference organized by Asia-Pacific Signal and Information Processing Association (APSIPA 2012)*, pp. 1–4, December 2012.

Other Publications (All in Japanese)

Technical Reports

17) Naoyuki Kanda, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno, "Spoken Dialogue System for Database Retrieval using Contextual Constraint," in *JSAI Technical Report*, Vol.41, SIG-SLUD-A401-4, pp. 21–26, June 2004.

18) Naoyuki Kanda, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno, "Experimental Evaluation of Spoken Dialogue System using Contextual Constraint in

Database Retrieval Task," in *IPSJ SIG Technical Report*, Vol. 2005, No. 12 (2005-SLP-55), pp. 107–112, Feburary 2005.

19) Naoyuki Kanda, Kazunori Komatani, Mikio Nakano, Kazuhiro Nakadai, Hiroshi Tsujino, Tetsuya Ogata, and Hiroshi G. Okuno, "Robust Domain Selection using Dialogue History in Multi-Domain Spoken Dialogue System," in *IPSJ SIG Technical Report*, Vol. 2006, No. 12 (2006-SLP-60), pp. 55–60, Feburary 2006.

20) Naoyuki Kanda, Takashi Sumiyoshi, Masahito Togami, and Yasunari Obuchi, "Evaluation of multistage rescoring strategy for open-vocabulary spoken term detection," in *Proc. 2nd Annual Spoken Document Processing Workshop*, pp. 73–78, Feburary 2008.

21) Naoyuki Kanda, Ryu Takeda, and Yasunari Obuchi, "Fundamental Evaluation of Japanese Speech Recognition based on Deep Neural Networks," in *IPSJ SIG Technical Report*, Vol. 2013, No. 8 (2013-SLP-97), pp. 1–6, July 2013.

22) Yasunari Obuchi, and Naoyuki Kanda, "Practical Applications of Spoken Term Detection: Current Status and Issues," in *IPSJ SIG Technical Report*, Vol. 2011, No. 5 (2011-SLP-88), pp. 1–4, October 2011.

23) Yasunari Obuchi, Ryu Takeda, and Naoyuki Kanda, "Voice activity detection under noisy environment based on augmented execution of statistical noise suppression," in *IPSJ SIG Technical Report*, Vol. 2012, No. 18 (2012-SLP-94), pp. 1–6, December 2012.

National Convention Papers

24) Naoyuki Kanda, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno, "Efficient Dialogue Management in Spoken Dialogue Systems Using Contextual Structures," in *Proc. 66th IPSJ National Convention*, 4T-5, March 2004.

25) Naoyuki Kanda, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno, "Empirical Evaluation of Spoken Dialogue System using Contextual Constraint in Database Retrieval Task," in *Proc. 67th IPSJ National Convention*, 4R-7, March 2005.

26) Naoyuki Kanda, Kazunori Komatani, Mikio Nakano, Kazuhiro Nakadai, Hiroshi Tsujino, Tetsuya Ogata, and Hiroshi G. Okuno, "Robust Domain Selection using Dialogue History in Multi-Domain Spoken Dialogue Systems," in *Proc. 68th IPSJ National Convention*, 5M-1, March 2006.

27) Naoyuki Kanda, Takashi Sumiyoshi, Masahito Togami, and Yasunari Obuchi, "Spoken Utterance Retrieval based on Multistage Rescoring," in *Proc. of the 2007 Autumn Meeting of ASJ*, 3-Q-20, September 2007.

28) Masaki Ono, Takeshi Homma, Naoyuki Kanda, Kenji Nagamatsu, and Yukiko Nakano, "Spoken Dialogues via Speech Recognition and Natural Language Understanding Systems: The First Report for the Corpus Collection and an Analysis," in *Proc. 23th JSAI National Convention*, 1T1-1, June 2009.

29) Ryu Takeda, Naoyuki Kanda, and Yasunari Obuchi, "Distributed Search of Audio Keyword by Multi-stage Scoring using Document Score," in *Proc. of the 2013 Spring Meeting of ASJ*, 3-9-8, March 2013.